



TITLE:

画像処理のための不連続な解に対処する正則化手法(Dissertation_全文)

AUTHOR(S):

天野, 晃

CITATION:

天野, 晃. 画像処理のための不連続な解に対処する正則化手法. 京都大学, 1996, 博士(工学)

ISSUE DATE:

1996-01-23

URL:

<https://doi.org/10.11501/3108348>

RIGHT:

新 制
工
1016
京大附図

画像処理のための不連続な解に 対処する正則化手法

天野 晃

1995 年 9 月

画像処理のための不連続な解に 対処する正則化手法

天野 晃

1995 年 9 月

内容梗概

様々な機械の自動化に対する需要の増加に伴い，人間の視覚機能を代行する画像認識技術に対する要求は大きくなっている。

人間の視覚機構は，網膜を通して得られた輝度分布に対して，はじめに初期視覚と呼ばれる，画像から直接計算出来る特徴量を求める処理によって，初期スケッチと呼ばれる画像特徴に変換され，次に，これらの特徴量が統合されて，もとの三次元空間に関する記述を得る，という過程によって構成されていると考えられている．画像認識システムにおいても同様の構成を用いることが多いが，安定した認識結果を得るためには，ノイズや認識対象の変化に対して安定した結果を与える初期視覚の計算アルゴリズムが必要である．しかしながら，多くの初期視覚計算手法はノイズの影響を受けやすく，また認識する対象の性質によって結果の安定性が大きく変化する．

初期視覚の問題は，三次元空間の情報を，二次元に投影された画像から推定する問題であるから，データのみからでは解が決定できない不良設定問題であることが多い．不良設定問題に対して安定した解を与える手法として正則化手法がある．

正則化手法は，数学的に厳密に定義された正則化理論に基づいた手法であり，不良設定問題に対して，得られる解の性質，ノイズに対する影響などが明確であるという特長を持つ．このため初期視覚問題に対しても，広く応用されている．

正則化手法を適用する際には，問題を詳しく検討し，用いる関数

やパラメータなどを決定しなければならない。

正則化手法では解に関する一般的な制約を設けることによって、安定な解を推定するが、従来の適用例では、Tikhonov 型の関数と呼ばれる、連続解でかつ解の分布が滑らかであることを制約とする関数を用いる方法が多かった。しかしながら、多くの初期視覚の問題は不連続解を解として持つため、必ずしも理論的に妥当な解が得られているとはいえない場合が多かった。

不連続解を解に持つ問題に対して、不連続正則化手法が研究されている。しかしながら、不連続部分を確率的に決定する確率的正則化手法では、計算時間が大きくなり現実的な問題に対して解が得られないという問題がある、あるいは多価関数を用いる方法では不連続部分の解の挙動が不安定になる、GNC 法では不連続部分のモデルの制約がきついなどの問題がある。

また、正則化手法においては、一般的に与えられる解の性質と観測されたノイズを含むデータとのバランスを決める正則化パラメータと呼ばれるパラメータを決定しなければならない。従来の研究では、このパラメータは経験的に決められていたが、実際の応用においては、問題の解析によってパラメータを決定する手法が必要である。

さらに、正則化手法において解を計算するときには、解の初期値を与える必要がある。問題の性質によっては、初期値によって最適な解を得るのに必要な計算時間が大きく影響されることがある。従って、安定して良い初期値を与える手法も重要である。

第3章では、不連続解を含む解を推定する問題および、正則化パラメータの決定手法として、滑らかでない輪郭を検出する輪郭検出手法である、サンプル輪郭モデルを利用した Snakes について述べる。

連続解であることと解の分布が滑らかであることを制約とする正則化手法を応用した輪郭検出手法である、Snakes は、画像の局所的

性質だけでなく大局的な性質も考慮することから、局所的なノイズの影響を受け難い手法である。このような性質を保持したまま、滑らかでない部分が含まれる輪郭を検出できるように Snakes を拡張する。このため、事前に検出したい輪郭と類似の画像に対して輪郭を入力し、これをサンプル輪郭モデルとする。また、従来経験的にしか決定することができなかった、画像の輝度分布に関する制約と輪郭形状に関する制約のバランスを決定するパラメータを、2枚の画像と対応する輪郭を与えることによって、決定する手法について述べる。

第4章では、特定の運動を行っている領域のみが存在する画像からオプティカルフローを推定する手法によって、不連続解を含む解を推定する手法及び、安定した初期値を与える手法について述べる。

オプティカルフローの推定に、Tikhonov 型の安定化汎関数による正則化手法を用いた方法としては、Horn らの方法が挙げられる。この手法は、局所的なノイズの影響を受け難いという特長があるが、背景の前を対象物が通過している場合などの、オプティカルフローの分布に不連続部分が存在する場合に、良い結果が得られないという問題があった。この問題に対処するため、エッジをもとにオプティカルフローを推定する手法が提案されているが、エッジの検出精度に問題があり、必ずしも良い結果を与えない。これに対して本稿では、移動している対象物の運動の種類を限定することにより、解の分布に不連続部分を許す、特定の値のみを取る関数を制約とする不連続正則化手法を用いたオプティカルフロー推定法について述べる。また、本手法では、対象物の運動の初期値が必要であるが、ここでは勾配法にもとづく拘束線の投票により初期値を求める手法を提案している。特に、従来あまり検討されなかった、画像に含まれる空間周波数と量子化 bit 数による誤差を詳しく解析することによって、本手法が適用できる範囲を明らかにしている。

第 5 章では，不連続正則化手法の一つである GNC 法を 2 次元に拡張し，距離画像に適用することによって，ノイズを含む距離画像から曲面の推定を行い，推定した曲面に対してコンピュータグラフィックスで表示することを目的とした三次元形状モデルの生成を行っている．

距離画像からの曲面復元の問題は，主にコンピュータグラフィックスで表示することを目的とした研究が行われている．本論文では，2 次元の距離画像から曲面を推定する手法として，扱える信号の範囲が明らかな GNC 法を 2 次元に拡張したものを提案する．また，推定された曲面に対して，実時間での表示，および精密な表示の二つの目的に対応した形状モデルを生成出来る三次元形状モデル生成手法について述べる．本手法では，生成されるモデルが誤差に対して最適化されているので，誤差の少ないモデルを生成するという目的に対しては，最適な手法であると言える．

本論文で述べた手法によって，従来不連続解を解として含む問題に対して適用が困難であった正則化手法を，より広い範囲に適用することを可能にした．また，正則化手法を適用する際に問題になる，正則化パラメータに関しても，従来経験的に決定されていたため，決定した値に根拠がなく，得られる解にも明確な理論的根拠を与え難かったが，決定法を提案することによって，解の妥当性を与えることができた．さらに，解の計算に必要な初期値の計算手法について，問題に即した初期値計算法を用いることによって安定して解を得ることができることを示した．

目次

1	はじめに	1
2	初期視覚問題と正則化	3
2.1	不良設定問題	3
2.2	不良設定問題としての画像認識	4
2.3	不良設定問題の解法	5
2.4	正則化手法	6
2.5	初期視覚問題における正則化手法の応用とその問題点	8
2.5.1	Snakes による輪郭検出	8
2.5.2	オプティカルフロー推定	10
2.5.3	曲面復元	11
2.6	正則化手法応用の問題点	14
3	サンプル輪郭モデルを利用した Snakes	16
3.1	まえがき	16
3.2	Snakes の問題点	18
3.2.1	Snakes のエネルギー関数	18
3.2.2	正則化理論における関数の制約	19
3.2.3	Snakes のエネルギー関数の問題点	19
3.3	サンプル輪郭モデルを利用した Snakes	21
3.3.1	サンプル輪郭モデル	21

3.3.2	エネルギー関数	23
3.3.3	パラメータの計算法	25
3.4	実験	28
3.4.1	ノイズの少ない画像の例	30
3.4.2	ノイズの多い動画像の例	33
3.4.3	複雑な形状の例	34
3.5	むすび	36
4	不連続正則化手法を用いた並行移動物体のオプティカルフロー推定	39
4.1	まえがき	39
4.2	不連続部を含むオプティカルフローの推定	42
4.2.1	オプティカルフロー	42
4.2.2	極小値が全て 0 になる関数を安定化汎関数として用いた正則化	42
4.2.3	オプティカルフロー計算	44
4.3	勾配法を用いた並行移動領域のオプティカルフローの初期値推定	46
4.3.1	勾配法における拘束線	46
4.3.2	差分による輝度勾配の計算誤差	47
4.3.3	量子化誤差	51
4.3.4	輝度勾配の誤差と拘束線の変位	51
4.3.5	拘束線の変位を考慮した初期値計算	52
4.4	実験結果	61
4.4.1	勾配法による初期値推定	61
4.4.2	不連続部を含むオプティカルフローの推定	66
4.5	むすび	69

5	再帰的分割を用いた適応格子による距離画像からの三次元形状モデルの生成	73
5.1	まえがき	73
5.2	GNC 法を用いた距離画像からの曲面復元	76
5.3	三次元形状モデル	77
5.3.1	三次元形状モデルの表現方法	77
5.3.2	面パッチの形	78
5.3.3	双一次曲面	80
5.4	三次元形状モデルの生成	81
5.4.1	距離画像からの三次元形状モデルの生成	81
5.4.2	誤差関数	83
5.4.3	面パッチの形状変化	84
5.4.4	面パッチの再分割	87
5.4.5	面パッチの形状変化及び再分割	88
5.5	実験	90
5.5.1	面パッチの形状変化による形状モデルの生成	90
5.5.2	面パッチの形状変化及びその再分割による形状モデルの生成	92
5.5.3	実距離データからの形状モデル生成	97
5.6	むすび	102
6	まとめ	104

1 章

はじめに

様々な機械の自動化に対する需要の増加に伴い，人間の視覚機能を代行する画像認識技術に対する要求は大きくなっている。

人間の視覚機構は，網膜を通して得られた輝度分布に対して，はじめに初期視覚と呼ばれる，画像から直接計算出来る特徴量を求める処理によって，初期スケッチと呼ばれる画像特徴に変換され，次に，これらの特徴量が統合されて，もとの三次元空間に関する記述を得る，という過程によって構成されていると考えられている．計算機による画像認識も同様の過程によって実現出来ると考えられ，研究がすすめられてきた [1]．

画像認識システムの実現のためには，安定して誤りの少ない初期視覚特徴量を計算する手法が重要である．しかしながら，初期に行われた初期視覚の研究における成果は，環境変動に対して弱く，誤りの多い結果しか得られない手法が多かった．このような状況で，ノイズにも強く大域的な情報を扱う方法として正則化手法が使われるようになった．

正則化手法は，Tikhonov によって不良設定問題に対する安定した解を推定する手法として提案された [2]．画像認識における初期視覚の問題への応用としては Poggio らによるもの [3] があげられるが，

あまり理論的な検討を加えなくても比較的良い結果を与えることから、その後行われた応用は、かならずしも理論的に正当な応用とはいえないものも多い。

本論文では初期視覚の問題に対して、正則化理論に則り、問題を再検討することによって、理論的に明確な解を得る方法について議論する。

第2章では、正則化理論について説明し、初期視覚における種々の問題との関連について説明する。第3章では、輪郭検出において、滑らかでない輪郭形状を検出する手法として、サンプル輪郭モデルを利用した Snakes について説明する。第4章では、オプティカルフロー検出において、不連続なフロー変化に対処する手法について説明する。第5章では、不連続な面に対する距離画像から、ノイズ除去を行い、これに対して、三次元形状モデルを生成する手法について述べる。最後に、第6章でまとめを述べる。

2 章

初期視覚問題と正則化

2.1 不良設定問題

良設定問題という概念は、物理の問題において、どのクラスの問題が微分方程式によって解けるかを明らかにするため、J.Hadamardによって導入された。

定量的に定式化された問題は、観測量 u から解 $z : z = R(u)$ を求める問題と考えられる。ここで、 u, z は測度空間 U, F に属する要素であり、測度空間 U, F はそれぞれ $\rho_U(u_1, u_2)$ (ただし $u_1, u_2 \in U$)、 $\rho_F(z_1, z_2)$ (ただし $z_1, z_2 \in F$) を測度として持つ。なお、通常測度は問題によって決まる。

ここで、全ての $u \in U$ に対して空間 F の中に一意な解 $z = R(u)$ が存在するとする。 $u_1, u_2 \in U$ で $z_1 = R(u_1), z_2 = R(u_2)$ また、 $z_1, z_2 \in F$ であるとする。 全ての正の数 ϵ に対して $\rho_U(u_1, u_2) \leq \delta(\epsilon), \rho_F(z_1, z_2) \leq \epsilon$ であるような $\delta(\epsilon)$ が存在するとき、この問題は空間 (F, U) において安定 (stable) であると言う。

良設定問題とは以下のような条件を満たすような問題である。

1. 全ての $u \in U$ に対して空間 F の中に解 z が存在する
2. 解は一意である

3. 問題は空間 (F, U) において安定である

ここで注意しなければならないのは、問題が良設定であるかどうかは、空間 F, U の測度 ρ_F, ρ_U に依存するということである。

従来、良設定問題のみが数学的に扱われてきた。しかしながら、上記の条件を満たさない不良設定問題は多く、これらの問題を解く手法が必要である。

2.2 不良設定問題としての画像認識

画像を入力として何らかの処理を行う場合、三次元空間を二次元平面に写像した画像から、もとの三次元空間の情報を復元する必要があることがある。このような問題は、一般に解に一意性がない不良設定問題となる。

ところで、1970 年代に Marr によって提唱された、画像認識のパラダイムでは、画像認識は以下のような過程を経て達成される。

1. 画像から直接計算出来る特徴量を求める。これは初期視覚問題と呼ばれる。
2. 初期視覚問題によって得られた特徴量を統合し、三次元情報を復元する。

このような過程で画像認識を行う場合、三次元情報を復元する処理にとって有用な、初期視覚問題の解を得るアルゴリズムが必要となってくる。

初期視覚の問題としては、認識対象となる物体の物理量に直接関係する、画像上の直線検出、輪郭検出、領域分割、ステレオ画像からの距離情報復元、オプティカルフローの検出、陰影からの形状復元等があげられる。これらの問題はいずれも画像情報のみでは解が一意

に定まらない不良設定問題である。従って、問題に対する制約を用いないと解が定まらないため、制約を与えて問題の解を得る方法が必要である。

2.3 不良設定問題の解法

z と u の間に $Az = u$ という関係があり、右辺の値が誤差 δ でわかっているとする。つまり、正確な値である u_T ではなく、 $\rho_U(u_T, \tilde{u}) \leq \delta$ であるような \tilde{u} が与えられているとする。

このような場合、 z を求めるために、 $\rho_U(Az, \tilde{u}) \leq \delta$ となるような z の集合 Q_δ から解を探索するのは自然な方法である。しかしながら、このような集合は場合によって大き過ぎ、それぞれの要素がかけはなれたところに存在するようなことがある。

従って、このような集合から、解を探索する空間を選択する方法が必要になる。このような選択方法として、選択法と正則化手法が知られている [2]。

• 選択法

選択法においては、新たに解空間 M を導入し、以下のような z_0 を選ぶことによって解を決定する。

$$\rho_U(Az_0, u) = \inf_{z \in M} \rho_U(Az, u)$$

ここで、空間 M は閉集合でなければならず、さらに真の解を含まなければならない。

選択法は、一般に解空間 M を生成するのが困難であるが、 M の近似的な空間を生成することが可能な問題に対しては、良い解を与える手法である。

- 正則化手法

正則化手法では、オペレータ R を定義し、観測データ u とその誤差 δ から解 $z = R(u, \delta)$ を決定する。正則化手法については次節で詳しく説明する。

画像認識における初期視覚の問題では、解空間 M を生成する困難さから選択法は用いられないが、正則化手法は適用が容易であり、比較的良い結果が得られることから、広く応用されている。

2.4 正則化手法

いま、未知量 z が、線形操作 A を受けた観測量 u として観測されるとする。

$$Az = u$$

真の解 $z_T \in F$ と真の観測量 $u_T \in U$ は $Az_T = u_T$ と表されるとすると、以下のような条件を満たすとき $R(u, \alpha)$ は正則化オペレータであるという。

1. $R(u, \alpha)$ は $\alpha > 0$ で、かつ

$$\rho_U(u, u_T) \leq \delta \leq \delta_1$$

であるような u に対して定義される。

2. 全ての $\epsilon > 0$ に対して、 $\delta(\epsilon) \leq \delta_1$ となる $\delta(\epsilon)$ が存在し、以下の条件を満たす $\alpha = \alpha(\delta)$ が存在する。

$$\rho_U(u_T, u_\delta) \leq \delta(\epsilon)$$

ならば,

$$\rho_F(z_T, z_{\alpha(\delta)}) \leq \epsilon$$

である. ただし, $z_{\alpha(\delta)} = R(u_\delta, \alpha(\delta))$ である.

このとき, $z_\alpha = R(u_\delta, \alpha)$ によって得られる解を正則化解と呼ぶ.

ここで, $\rho_U(u_T, u_\delta) \leq \delta$ であることがわかっているとすると, 正則化オペレータの定義より $\alpha = \alpha(\delta)$ を以下のように選ぶことが出来る. すなわち, $\delta \rightarrow 0$ となるときに $z_\alpha = R(u_\delta, \alpha(\delta))$ は, 真の解 z_T に近づく. つまり $\rho_F(z_T, z_{\alpha(\delta)}) \rightarrow 0$ となる.

正則化オペレータの具体的な例として安定化汎関数を用いる手法があげられる. これは以下の条件を満たす汎関数を利用する手法である.

1. 空間 F の部分空間 F_1 上で定義される非負の汎関数を $\Omega[z]$ とする. (この汎関数を安定化汎関数と呼ぶ)
2. z_T は $\Omega[z]$ の定義域に含まれる.
3. 全ての正の数 d に対して, $\Omega[z] \leq d$ となる空間 F_1 の要素 z で構成される部分空間は, コンパクトである.

これらの条件を満たす関数は, d を閾値としたときに, $\Omega[z] \leq d$ となる空間がふたつ以上の部分に分離しない関数である. このような関数として, Tikhonov 型の安定化汎関数が知られている. Tikhonov 型の安定化汎関数を用いた正則化オペレータは, 以下の関数の最小化によって得られる値を解とすることにより得られる.

$$E = \|Az - u\|^2 + \lambda \|Pz\|^2 \quad (1)$$

$$Pz = \sum_{r=0}^p q_r(x) \left(\frac{d^r z}{dx^r} \right)^2 dx$$

ただし、 $q_r(x)$ は非負の連続関数である。

ここで、式 (1) における右辺第 1 項は、観測量との差を表すことから、ペナルティ関数と呼ばれている。

2.5 初期視覚問題における正則化手法の応用とその問題点

初期視覚問題に対する解法として、性質が明らかな Tikhonov 型の安定化汎関数を利用した手法が研究されてきた [3]。Tikhonov 型の安定化汎関数は、解の分布の連続性を制約とするため、対象となる画像中で解が連続的に変化する特徴量を得るときにしか用いることが出来ない。しかしながら、実際には解が不連続に変化する場面に対しても Tikhonov 型の関数が用いられることが多く、良い解が得られない原因となっている。

ここでは、正則化手法が応用されてきた初期視覚問題を取り上げ、その問題点とともに示す。

2.5.1 Snakes による輪郭検出

Kass ら [5] によって提案された動的輪郭モデル (Active Contour Model) Snakes は、画像の大局的な情報を利用し、輪郭検出や主観的輪郭の検出を行うことができる手法であり、ノイズの影響を受けにくく、欠落した情報をも補って輪郭を検出できる。その原理は画像の濃度に関する制約と対象物の形状に関する制約との組み合わせである。

Snakes では、検出したい輪郭の濃淡情報に関する一般的な性質を

表現する外部エネルギーと呼ばれる関数と、検出したい輪郭形状に関する一般的な性質を表現する内部エネルギーと呼ばれる関数の線形和を最小化することによって輪郭検出を行う。

通常外部エネルギーとしては、以下のものを用いる。

$$E_{image} = -|\nabla I(v(s))|^2$$

なお、 $I(x, y)$ は画像の輝度を表し、 $v(s)$ ($0 \leq s < 1$) は輪郭を表す。内部エネルギーとしては、輪郭形状の滑らかさを表す以下の関数を用いることが多い。

$$E_{int} = \alpha |v_s(s)|^2 + \beta |v_{ss}(s)|^2$$

$v_s(s)$ は輪郭が収縮する方向に移動すると値が小さくなるため、膜のような性質を表す関数といえる。また、 $v_{ss}(s)$ は輪郭形状が円に近づく程値が小さくなるため、板バネのような性質を表す関数といえる。 α および β はそれぞれの性質の重みを決める定数である。Snakes では、これらの関数を用いて以下の関数を最小化することにより、輪郭を検出する。

$$E_{snakes} = \int_0^1 (E_{int} + w_{image} E_{image}) ds$$

ここで w_{image} は二つの制約のバランスを決めるパラメータである。

Snakes では、滑らかな形状の輪郭を検出することを想定しているため、輪郭形状に滑らかでない部分が存在する場合、良い解が得られないという問題がある。

これに対処するために圧力エネルギーを導入した手法 [20] など、安定化汎関数を修正したものが挙げられるが、いずれも関数は連続解のみを解として含むものであるため、本質的には問題は解決されていない。

また、検出される輪郭は、正則化パラメータの値によって大きく変化する。正則化パラメータについては、従来の研究では、経験的に設定しているものがほとんどである。問題に対して適切な安定化汎関数を選択することにより、広い範囲の正則化パラメータの値で解を得ることが可能である。このことより、おおまかに決定した正則化パラメータを用いても、適切な安定化汎関数を用いていれば、ある程度良い解が得られるといえる。しかしながら、経験的に決定した正則化パラメータは、その妥当性に理論的根拠がないという問題がある。従って、正則化手法を適用する場合、正則化パラメータの決定手法も同時に検討することが重要である。

2.5.2 オプティカルフロー推定

運動している対象物の、画像上の見かけの運動を表すオプティカルフローは、計算機による画像認識のために有用な情報である。従来からオプティカルフローを推定する方法は研究されてきたが、この問題は画像情報のみを用いたのでは解が一意に定まらない不良設定問題であるため、種々の制約や知識を導入することが行われてきた。

移動している物体の輪郭を検出し、それらの対応を取る方法、あるいは、移動している物体の領域を検出し、それらの対応を取る方法、さらに、運動に関する不変量を計算し、それらのクラスタリング結果を用いる方法などが研究されているが、中でも、正則化手法を用いる Horn らの方法 [6] は、大局的な情報を利用し、ノイズの影響を受けにくいという特長を持つ方法として知られている。

Horn の方法では、以下のようにしてオプティカルフローの推定する。動画像の輝度を $I(x, y, t)$ とし、各画素に対応するオプティカルフローを (u, v) とすると、以下の関数を最小化することによってオプティカルフローを求めることが出来る。

$$\int \int \{(I_x u + I_y v + I_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)\} dx dy$$

正則化手法を用いる方法は、理論的根拠が明確であるため、アルゴリズムの能力が明らかであるという特長がある。

Horn の方法では、安定化汎関数として Tikhonov 型のものを用いているが、オプティカルフローが画像上で滑らかに変化するという制約を満たす場合には、この手法によって正しい解が得られる。しかしながら、この方法ではオプティカルフローが滑らかに変化するという制約を用いているため、背景の前に平行運動あるいは回転運動している対象物がある場合など、オプティカルフローが不連続に変化するような問題に対しては、良い解を与えないという問題がある。

これらの問題に対して、不連続解を含む解を扱う、不連続正則化手法がいくつか提案されている。しかしながら、これらの手法では事前に不連続部分に関する詳細な情報が必要であったり、扱える不連続関数が限られるという問題がある。

また、上式を最小にする解を数値的に計算する場合、関数が非常に複雑であるため一般に最適解を計算することは不可能であり、通常はある初期値から始めて、局所最適解を探索することになる。従って、初期値の選択によって解が変わることになる。一般的に良い初期値を得る計算法を実現することは非常に困難であるので、問題に対して他の手法等によって良い初期値を得ることも必要である。

2.5.3 曲面復元

距離画像やレーザー測距などで、観測者から対象物までの距離が与えられているときに、対象物の表面の曲面を再構成する問題は曲面復元の問題として従来から研究されている。

XY 平面上の曲面の高さが、離散的に $d(x, y)$ で与えられているとする。これらの点から復元する曲面を $f(x, y)$ とし、 f から曲面の高さをサンプリングする関数を S とすると、以下の関数を最小化することによって曲面の復元を行うことが出来る。

$$\int \int ((Sf - d)^2 + \lambda(f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2)) dx dy$$

この例では、安定化汎関数として Tikhonov 型のものが用いられているため、いたるところで滑らかな曲面でなければ良い解が得られない。Tikhonov 型の安定化汎関数は、この解空間に対して、滑らかな解という制約を満たす部分空間を与えるようなものである。これに対して、求めようとしている問題が解に不連続部分を含む場合、Tikhonov 型の安定化汎関数を用いることは、解空間に真の解を含まなければならないと言う安定化汎関数の制約から言っても誤った応用であると言える。

解の分布に不連続部分を含む問題に解を与える場合、不連続部分に関する何らかの情報が必要である。このような方法として提案されているものには、制御された連続性の拘束 (controlled-continuity constraint) を用いる方法、スプライン関数を用いる方法、ライン過程 (line process) を用いる方法、多価関数を用いる方法、GNC 法などがある。

制御された連続性の拘束を用いる方法 [9] では、解の中に不連続部分を示すフラグを設けることにより、解を得るが、この方法では不連続部分を事前に設定することが必要になるため、事前に不連続部分の情報が完全にわかっている必要がある。

また、スプライン関数を用いる方法 [10] では、解に関する拘束としてスプライン関数を用い、スムージング後の信号に閾値以上の不連続が検出されたところを不連続点としている。しかしながら、この手法では、傾きの不連続は表現できても値の不連続が表現できな

いため、値の不連続を含む問題に対しては適用できない。また、2次元への拡張が難しいという問題もある。

ライン過程を用いる方法 [11] では制御された連続性の拘束を用いる方法と同様に解の不連続に関する仮定を設け、これに関する確率分布を与えることにより確率的に解を推定するが、この方法では計算量が膨大になり実用時間では問題が解けないという問題がある。

多価関数を用いた方法 [12] では、連続な二つあるいはそれ以上の関数を多重解として持つ関数を拘束として用いることにより、解が不連続になる部分でも制約を表す関数の値としては連続的な変化をするような拘束を用いるが、この方法では二つの関数が既知である必要があり、さらにこれらの値が近くなる部分において解の挙動が不安定になるという問題がある。

GNC 法 [13] では、スプライン関数を用いる方法と同様に、スムージング後の信号に閾値以上の不連続が検出されたところを不連続点とする。

不連続部分が Z 個存在する、正規分布に従うノイズを含む 1 次元の観測データ d_i から、元のデータ u_i を推定する問題は以下の関数の最小化問題として定義出来る。

$$E = \sum_i ((u_i - d_i)^2 + \lambda^2(u_i - u_{i-1})^2) + \alpha Z \quad (2)$$

GNC 法による推定では、以下のようなことがいえる。

1. $d_i = \pm h/2$ であるとき、 $h > h_0 = \sqrt{2\alpha/\lambda}$ ならばその不連続を検出できる。
2. 傾きが $h_0/2\lambda$ であるような部分を不連続部分として検出する。

しかしながら、 d_i が分散 n_0^2 のノイズを含む時、不連続部分の精度には $8Ln_0^2/h^2$ の不確かさが残る。

GNC 法では, 上記の問題を解くために, 以下のような関数族を用いる.

$$F^{(p)} = \sum_i ((u_i - d_i)^2 + g^{(p)}(u_i - u_{i-1})) \quad (3)$$

$$g^{(p)}(t) = \begin{cases} \lambda^2 |t|^2 & \text{if } |t| < q \\ \alpha - c(|t| - r)^2/2 & \text{if } q \leq |t| < r \\ \alpha & \text{if } |t| \geq r \end{cases} \quad (4)$$

ただし,

$$c = \frac{1}{2p}, \quad r^2 = \alpha(\frac{2}{c} + \frac{1}{\lambda}), \quad q = \frac{\alpha}{\lambda^2 r}.$$

連続な関数である F^1 を最小化し, その解を用いて漸近的に, p を 0 に近づけていった F^p を最小化する解を求め, 最終的に F^0 を最小化する解を求める.

GNC 法は, 理論的な根拠が明確であり, 検出できる不連続点の性質や精度が明らかであるが, 1 次元信号を対象とした手法であり 2 次元画像に適用できないという問題がある.

2.6 正則化手法応用の問題点

従来の正則化手法応用の問題点をまとめると以下ようになる.

1. 解が連続で滑らかでない場合, 良い解が得られない
2. 正則化パラメータの決定法が必要
3. 数値計算の初期値計算法が必要

本論文ではこのような問題に対し，不連続部分に関してなるべく少ない仮定を設定するだけで，解を得る方法として，不連続解を解空間に持つ安定化汎関数を用いる手法について述べる．そして，対象となる問題のサンプルを提示することによって，パラメータを計算する手法，さらに問題の性質を利用して他の手法で初期値を計算する手法の利点について述べる．また，誤差を含む距離画像から GNC 法によって真の解を推定する手法，及びその推定した距離画像に対して，三次元形状モデルを生成する手法について説明する．

3 章

サンプル輪郭モデルを利用した Snakes

3.1 まえがき

画像認識のための初期視覚，つまり画像から直接得られる特徴量としては，輪郭や領域など様々なものがある．その中でも特定の領域を検出する処理は，その後の認識処理の処理要素となることから重要な処理であると言える．

これらの問題は，観測したデータのみからでは解が一意に定まらないという意味で不良設定問題である．正則化手法は，不良設定問題に対して解を得る手法として広く応用されているが，従来の応用で利用されてきた Tikhonov 型の安定化汎関数を用いたのでは，連続で滑らかな解を持つ問題に対してしか良い解を得ることができず，初期視覚問題で扱う必要性の高い，区分的に連続な解を解として含む問題に対しては必ずしも良い解が得られるとは限らないという問題があった．また，正則化手法では，観測されたデータと推定した解との差を表すペナルティ関数と推定した解の一般的な性質を表す安定化汎関数との比を決める正則化パラメータを決める必要があるが，このパラメータによって解が変化してしまうため，パラメータを経

験的に決定した場合、解の妥当性に関する理論的な根拠が不明確になるという問題があった。

本章では、これらの問題に対して、Kass ら [5] によって提案された、正則化手法の応用である、動的輪郭モデル (Active Contour Model) Snakes を拡張し、区分的に連続な輪郭の検出を行う。さらに、2 枚の画像とその画像における輪郭とを例示することによって正則化パラメータを計算する手法も提案する。

輪郭検出手法は従来から研究されている。一般化ハフ変換による任意の形状の検出 [14] では、ある既知の形状に対してその輪郭上の全ての点が同一の点に投票する投票空間を設定し、画像上の全ての輪郭候補に対して投票空間に投票することにより輪郭検出を行う。しかし、この手法では検出する輪郭の形状は既知である必要があり、また投票空間が必要であることから計算に要するメモリが膨大になるという問題がある。また同様の手法に Lamdan[15] らの Geometric Hashing 法が挙げられる。この手法では、既知の複数の形状に対して事前に局所特徴にもとづく不変空間を生成し、検出の段階では、検出対象のエッジをこの不変空間へ投票することにより、対象の識別と検出を同時に行っている。この手法では扱える輪郭の形状が複数のものに拡張されているが、対象の輪郭形状は既知である必要があり、幾何学的な形状以外の制約が既知である場合などの輪郭検出を行うことは不可能である。

Kass らによる Snakes は、画像の大局的な情報を利用し、輪郭検出や主観的輪郭の検出を行うことができる手法であり、ノイズの影響を受けにくく、欠落した情報をも補って輪郭を検出できる。Snakes は厳密な意味では正則化ではないので、エネルギー関数の最小点付近における収束性に問題がでてくる。これに対処するために弾性制約モデルを用いた手法 [16]、エッジモデルを領域を用いて定義した手

法 [17], 圧力エネルギーを導入した手法 [20] などがあるが, これらの手法は, エネルギー関数を問題に合わせて構成することによって収束性を良くしているため, 汎用性は低く, また収束性の問題も完全には解決はされていない.

本章では Snakes を正則化理論の考え方から捉え直し, Snakes におけるエネルギー関数が満たすべき条件を明確にする. その後に新たにサンプル輪郭モデルを利用した Snakes を提案し, これが画像に関する制約と形状に関する制約とのバランスを決めるパラメータの変動に対して安定性があることを実験により検証する. さらに, 2 枚の画像とその画像における輪郭とを提示することにより Snakes において適当に定めていたパラメータの計算を行う方法について述べる.

3.2 Snakes の問題点

3.2.1 Snakes のエネルギー関数

正則化理論の応用として, Kass[5] らによって提案された Snakes (以下 Snakes と呼ぶ) は, 画像のある輪郭 $v(s)$ 上に定義される内部エネルギーと画像エネルギーとの線形和であるエネルギー関数の最小化問題として定式化される.

$$E_{snakes}(v(s)) = \int \{E_{int}(v(s)) + w_{image}E_{image}(v(s))\} ds \quad (5)$$

ここで, E_{int} は輪郭の滑らかさを表す内部エネルギーであり, 形状に関するエネルギーと呼ばれる. これは, 輪郭 $v(s)$ の一次微分 $v_s(s)$, 及び二次微分 $v_{ss}(s)$ に関するエネルギーの和を用いることが多い.

$$E_{int}(v(s)) = \frac{1}{2} \{ \alpha |v_s(s)|^2 + \beta |v_{ss}(s)|^2 \} \quad (6)$$

α および β はそれぞれ重み係数である。 E_{int} は、輪郭の形状が円に近いほど、また輪郭の長さが短いほど、値が小さくなる。

E_{image} は画像エネルギーであり、画像濃度 $I(v(s))$ の一次微分により、次のように与えられる。

$$E_{image}(v(s)) = -|\nabla I(v(s))|^2 \quad (7)$$

この関数は輪郭上の画素における画像濃度の一次微分の和が大きいほど値が小さくなる。

3.2.2 正則化理論における関数の制約

ところで、正則化理論において、ペナルティ関数と安定化汎関数は、観測量 u が誤差を含まないときに、最適な推定量において値が最小になるような関数でなければならない。なぜならば、誤差のない入力を与えた時に、ペナルティ関数と安定化汎関数が最小になれば、バランスを決めるパラメータ λ によって最小値を与える点が変わるからである (図 1 参照)。

3.2.3 Snakes のエネルギー関数の問題点

正則化理論におけるペナルティ関数は、観測量と推定量の差を評価する関数であるが、Snakes のエネルギー関数 (式 (5)) においてこれに相当するものは、次に示す画像エネルギーの項である。

$$\int w_{image} E_{image}(v(s)) ds \quad (8)$$

また、安定化汎関数は、解に関する一般的な性質を表す関数であるから、Snakes においては、輪郭の形状によって決まるエネルギーの項がこれにあたる。

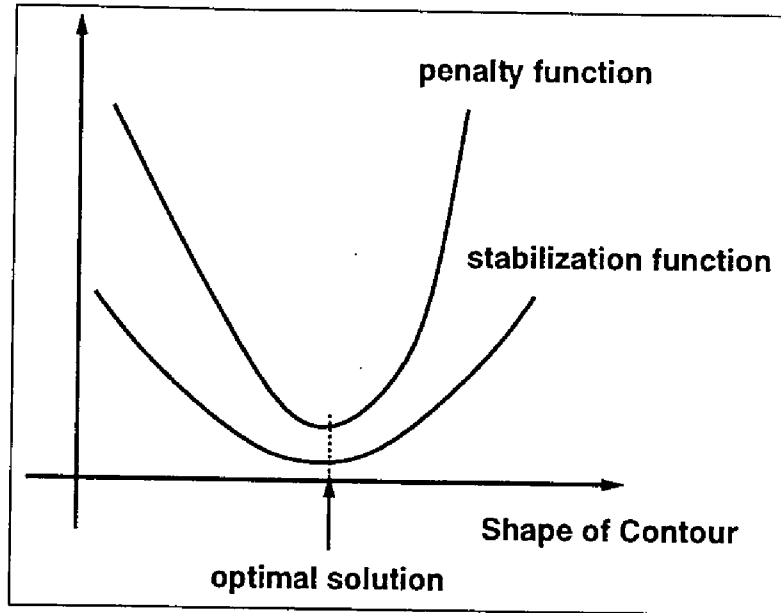


図 1: 正則化理論におけるエネルギー関数

Fig 1: Energy Functions in Regularization Theory

$$\int E_{int}(v(s))ds \quad (9)$$

ペナルティ関数 式 (7) が Snakes における画像エネルギー, つまりペナルティ関数である. このペナルティ関数は, 最適な推定量つまり $v(s)$ が目的とする輪郭である場合にも一般に最小にはならない (図 2 参照). これは, 必ずしも目的とする輪郭において, 輪郭上の画素における画像濃度の一次微分の自乗和が最大になるとは限らないからである. このような関数を用いた場合, 正則化理論における正則化パラメータ λ にあたるパラメータ w_{image} によって得られる解が変動するという問題が起こる.

安定化汎関数 Snakes における安定化汎関数は形状に関するエネルギー E_{int} であるが、これには式 (6) が用いられることが多い。この式の右辺第一項は輪郭に対して膜の性質を表現する関数、右辺第二項は輪郭に対してバネの性質を表現する関数である。また α, β はそれぞれ、膜の性質及びバネの性質の重要度を表す定数であり、いずれも輪郭の滑らかさに関係するものである。この関数は、与えられた輪郭が円形に近いほど値が小さくなる関数であり、 α, β を輪郭に沿った分布定数としても、任意の輪郭において最小になる関数を構成できるとは限らない。従って、Snakes を複雑な形状へ収束するように制御することは難しい。目的の輪郭において値が最小にならない関数を用いた場合、図 2 左図に示すように、パラメータ w_{image} の値によって、式 (5) を極小にする解が変動するという問題が起こる。

3.3 サンプル輪郭モデルを利用した Snakes

Snakes を利用する場合、従来の研究 [16] [17] [19] [20] [21] で用いられている式 (6) や、式 (7) を用いたのでは、必ずしも目的の輪郭において最小になるエネルギー関数が存在することを保証することができない。従って、一般に Snakes を目的とする輪郭に収束させることはできない。そこで、本研究では目的の輪郭においてエネルギー関数が最小になる、サンプル輪郭モデルを利用した Snakes (以下 M-Snakes と表記する) を提案する。

3.3.1 サンプル輪郭モデル

ある画像とその画像における任意の閉曲線 (一般的には特定の対象物の輪郭となるが、必ずしもこれが輪郭である必要はない) を合わせてサンプル輪郭モデルと呼び、その画像と輪郭をそれぞれモデル画像、モデル輪郭と呼ぶ。M-Snakes は、サンプル輪郭モデルにおい

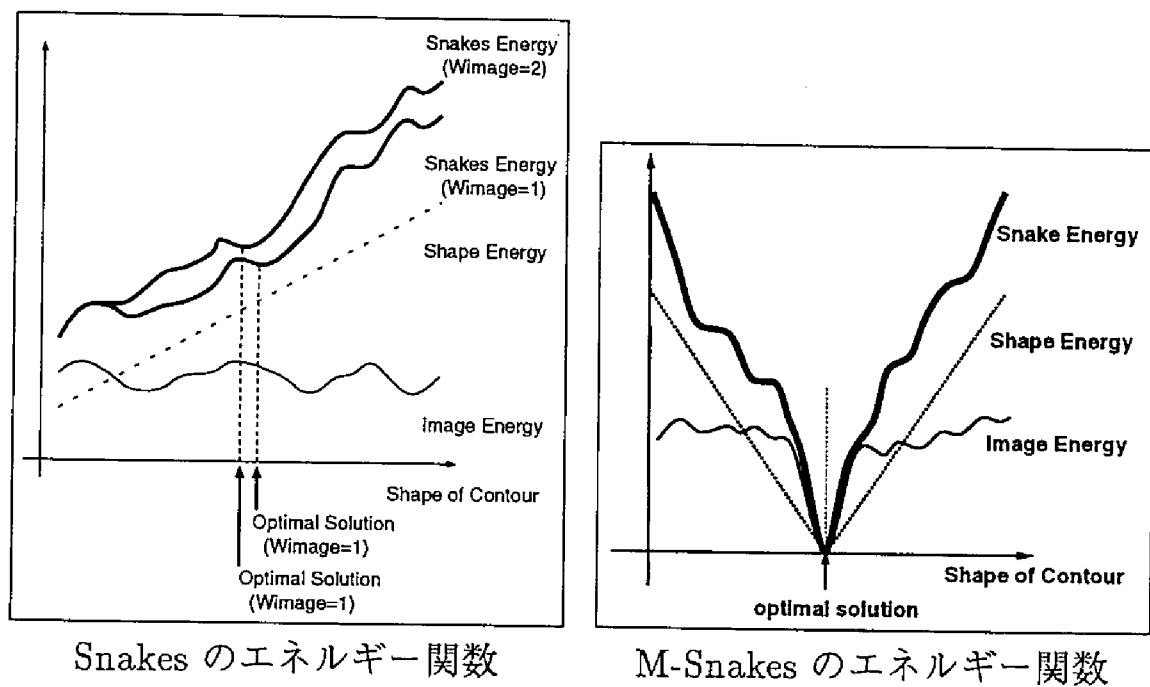


図 2: Snakes 及び, サンプル輪郭モデルを利用した Snakes におけるエネルギー関数

Fig 2: Energy Functions in Snakes and Snakes using Sample Contour Model

て、画像エネルギーと形状に関するエネルギーの両方の値が最小になるような関数をエネルギー関数とするような Snakes と定義する。従って、モデル画像に対して M-Snakes を適用すると、ふたつのエネルギーのバランスを決めるパラメータ w_{image} の影響を受けずモデル輪郭を処理結果として得ることができる。この M-Snakes は、時系列画像のように性質の同じ複数の画像から対象物の輪郭を検出するような場合に有効である。すなわち、時系列画像の一つの画像とそのなかの対象物の輪郭をサンプル輪郭モデルとすることにより、次のフレームの画像 (同様の性質を持つと考えられる) から同じ対象物の輪郭を検出することができると考えられる。

サンプル輪郭モデルとしては、どのような閉曲線を与えても良い。なぜなら、M-Snakes はそのエネルギー関数として、与えられた閉曲線の画像エネルギーとの差の絶対値を用いるため、その濃淡分布になるべく近い輪郭を検出しようとするからである。

3.3.2 エネルギー関数

モデル画像を I_0 , モデル輪郭を $v_0(s)$ とする。また、輪郭検出を行う画像を I_1 , その画像における目的の輪郭を $v_1(s)$ とする。画像 I_0 における形状に関するエネルギー、式 (6) の値を $E_{int}^0(v_0(s))$, 画像エネルギー、式 (7) の値を $E_{image}^0(v_0(s))$ とする。これらの値は、サンプル輪郭モデルに関する形状及び濃度分布に関する情報を保持していると考えられるので、これを利用して M-Snakes のエネルギー関数を構成する。

M-Snakes では、画像エネルギー及び形状に関するエネルギーとして、従来の画像エネルギー及び形状に関するエネルギーと上述の $E_{int}^0(v_0(s))$ 及び $E_{image}^0(v_0(s))$ との差の絶対値を用いる。式で表すと以下のようなになる。

$$\begin{aligned}
E_{msnake} = & \\
& \int \{ |E_{int}(v(s)) - E_{int}^0(v_0(s))| \\
& + w_{image} |E_{image}(v(s)) - E_{image}^0(v_0(s))| \} ds \quad (10)
\end{aligned}$$

なお, $E_{int}(v(s))$, $E_{image}(v(s))$ は画像 I_1 上の関数である.

このようなエネルギー関数を用いる M-Snakes は, サンプル輪郭モデルにおいて各エネルギー関数の値が最小になるという性質を持つ (図 2 参照).

モデル画像以外の画像に対してこの M-Snakes を適用した場合, 目的の輪郭におけるエネルギー関数の最小性は保証されないため, パラメータ w_{image} による解の変動が起こる. しかしながら, 目的の輪郭の形状や輪郭上の画素の画像濃度がサンプル輪郭モデルとあまり変わらない画像の場合は, このエネルギー関数の値は目的の輪郭において最小値に近くなると考えられる. M-Snakes のエネルギー関数は, 最小値付近のみで値が小さくなるため Snakes のエネルギー関数に比べて最小値付近の関数値の変化率が大きいと考えられるので, ふたつのエネルギー関数のバランスを決めるパラメータ w_{image} の変動に対する収束結果の安定性が増すことが期待できる.

このようなことから M-Snakes は Snakes において問題であった, パラメータに依存した収束結果の変動及び複雑な形状に収束しにくいという性質に対して良好な性質を持つことが期待できる.

ところで, M-Snakes は $v(s)$ と $v_0(s)$ の対応がとれていないと, エネルギー関数が最小にならない. ここでは, $v(s)$, $v_0(s)$ を重心からの角度で正規化することによってこの対応づけを行っている.

3.3.3 パラメータの計算法

前述のように M-Snakes は, Snakes に比べ原理的にパラメータ w_{image} の変動に対して安定である. しかし, 実際の問題に適用する場合は, w_{image} の計算法が問題になる. ここでは w_{image} を 2 枚の画像におけるサンプル輪郭モデルを用いて決定する方法を考える.

M-Snakes を, モデル画像に対して適用した場合, w_{image} は, 最適解において値が最小になる関数の係数であるから, 解が収束する位置, すなわち検出される輪郭の位置や形状には影響を与えない. ところが, M-Snakes を他の画像に対して適用した場合, 目的の輪郭においては形状に関するエネルギー $|E_{int} - E_{int}^0|$ が最小にならないので w_{image} が収束結果に影響を与えるようになる.

用いる 2 枚の画像を I_0, I_1 とし, それぞれの画像における目的の輪郭を $v_0(s), v_1(s)$ とする. またそれぞれの画像における画像エネルギーを E_{image}^0, E_{image}^1 , 形状に関するエネルギーを E_{int}^0, E_{int}^1 とする. ここで, 画像 I_0 と輪郭 $v_0(s)$ をサンプル輪郭モデルとするような Snakes において, 画像 I_1 から輪郭 $v_1(s)$ を検出するような問題を考える. なお, $v_1(s)$ は, $|E_{image}^1 - E_{image}^0|$ の値が最小になる輪郭であると仮定する (図 3 参照).

w_{image} は, 形状に関するエネルギーと画像エネルギーの比を与えるパラメータである. エネルギー関数が $v_1(s)$ において最小になるためには図 4 に示す様に, $v_1(s)$ における形状に関するエネルギーの値を $\delta_{E_{int}}$, $v_0(s)$ における画像エネルギーを $\delta_{E_{image}}$ とすると, w_{image} は以下のような条件を満たさなければならない.

$$\begin{aligned} \delta_{E_{int}} &< w_{image} \delta_{E_{image}} \\ w_{image} &> \frac{\delta_{E_{int}}}{\delta_{E_{image}}} \end{aligned} \quad (11)$$

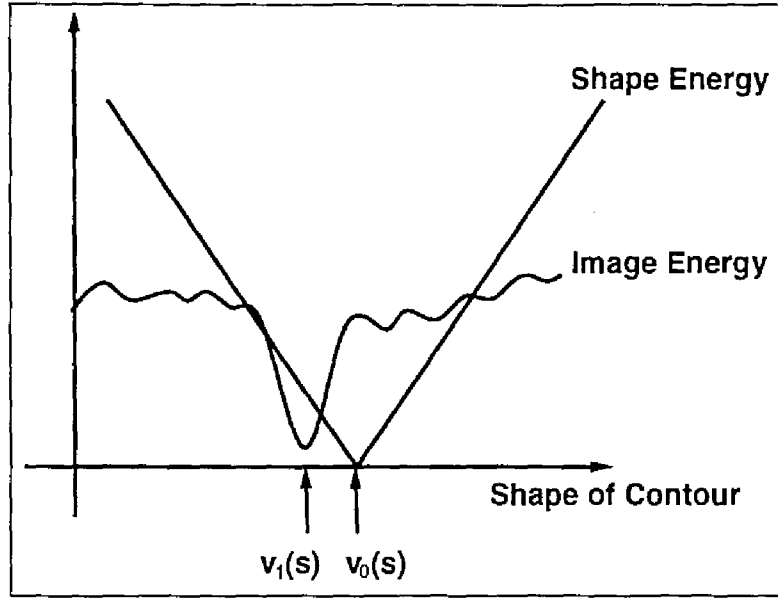


図 3: サンプル輪郭モデル以外の画像に適用した時のエネルギー関数

Fig 3: Energy Function when applied to Image other than Sample Contour Model

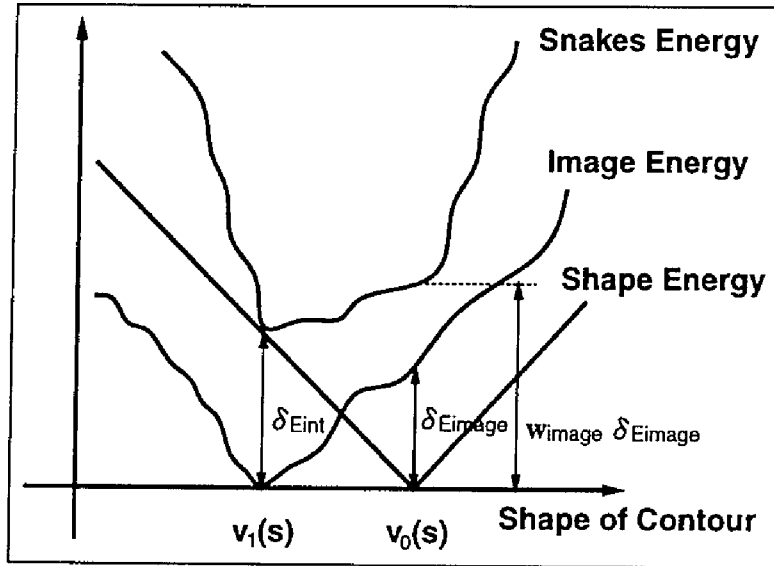
なお、 $v_1(s)$ における画像エネルギーの値は 0 にはならないことがあるが、サンプル輪郭モデルにおける画像と同様の性質を持つ画像に対して適用する場合その値は非常に小さくなると考えられるので、ここでは 0 と近似して計算を行った。

パラメータの決定に関係するのは、上式からわかるように δE_{image} と δE_{int} である。それぞれは画像及び形状の大局的な性質を与えるものとして以下のものを用いる。

$$\delta E_{image} = \int |E_{image}^1(v_1(s)) - E_{image}^0(v_0(s))| ds \quad (12)$$

$$\delta E_{int} = \int |E_{int}^1(v_1(s)) - E_{int}^0(v_0(s))| ds \quad (13)$$

次に w_{image} の上限について考える。画像エネルギーは目的の輪郭の近傍に極小値を持つことが多い。M-Snakes がここに収束しないた

図 4: w_{image} の下限Fig 4: Lower Bound of w_{image}

めには，式 (10) が解の近傍で極小値をもたない関数となる必要がある．このため， w_{image} を小さくし，解の付近で単調関数である形状に関するエネルギーの影響を強くする．しかしながら，この問題は多変数関数の極小値問題になるので，解析的に解くことは非常に困難である．そこで，ここでは近似的に計算する方法を考える．

上記の極小値は，輪郭の初期値と目的の輪郭との間にある画像濃度の一次微分の値が大きい点を輪郭が通る場合であると考えられる．ところで，Snakes の輪郭は，輪郭を構成する各点が，その点の近傍で決まるエネルギー関数の値の小さい点に移動することによって収束していく．従って，目的の輪郭に収束するためには，画像濃度の一次微分の値が大きいすべての点において画像エネルギーが形状に関するエネルギーよりも小さくならなければならない．そのためには，輪郭の初期値と目的の輪郭との間の画像濃度の一次微分が最大の点 p

に収束しない様に w_{image} を決定すればよい. この点における E_{image} の値 $E_{image}(p)$ は簡単に計算できるが, E_{int} の値 $E_{int}(p)$ は輪郭の形状を仮定しなければ求められない. そこでここでは, この点を通り前後は目的の輪郭にあり, 輪郭の重心からの動経の角度が $2\pi/N$ (N は Snakes を構成する点の数) 違うような輪郭を仮定し, $E_{int}(p)$ を計算する. (図 5 参照) この値を用いて表すと w_{image} は以下のような条件を満たさなければならない.

$$\begin{aligned} E_{int}(p) &> w_{image} E_{image}(p) \\ w_{image} &< \frac{E_{int}(p)}{E_{image}(p)} \end{aligned} \quad (14)$$

ゆえに

$$\frac{\delta E_{int}}{\delta E_{image}} < w_{image} < \frac{E_{int}(p)}{E_{image}(p)} \quad (15)$$

本節で計算されたパラメータの範囲は, M-Snakes が目的の輪郭に収束するための必要な条件であるが, 十分条件とはなっていない. 従って実際の問題に対しては, 計算された範囲でパラメータを調整する必要がある.

3.4 実験

本節では, 提案したサンプル輪郭モデルを利用した Snakes を用いて行った実験について述べる. なお, 実験で用いた Snakes は式 (10) であり, ペナルティ関数として, 画像濃度の一次微分 $\nabla I(v(s))$ を用いている. 実験では, 画像エネルギーと形状に関するエネルギーのバランスを決めるパラメータ, 式 (10) における w_{image} , を変化させることによって収束の安定性を検証した. また, 前節のパラメー

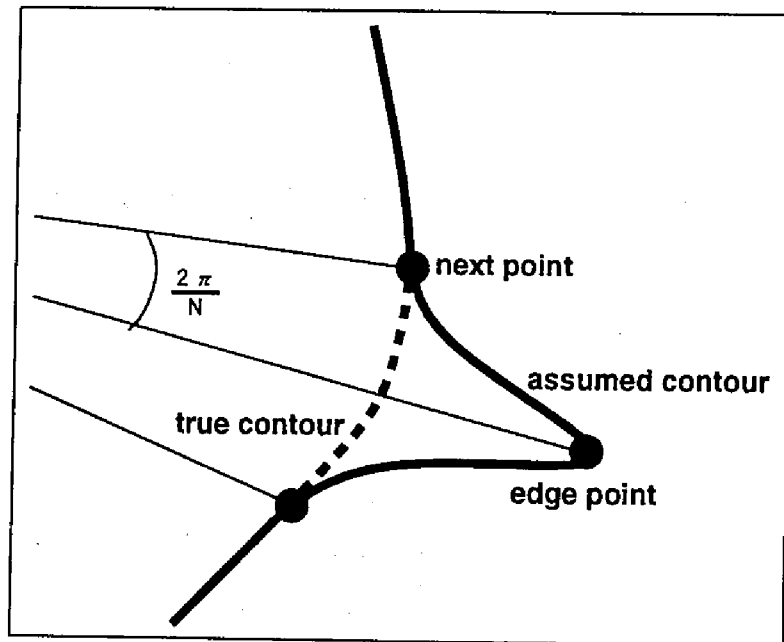


図 5: 仮定した輪郭

Fig 5: assumed contour

タ計算法を用いて行ったパラメータの値も示した。なお、パラメータ α, β は固定とし、それぞれ経験的に求められた値 0.8, 0.5 を用いた。また、収束計算は Amini[23] の方法による動的計画法を用いた。実験に用いた画像はノイズの少ない画像の例、およびノイズの多い動画像の例では 512×512 画素、複雑な形状の例では 320×480 画素で、それぞれ 1 画素あたり 256 階調の濃淡画像である。なお、本章で掲げる図には画像の中の必要な部分のみを示した。また、輪郭を構成している点の数は 64 点である。実験では w_{image} の範囲は有効数字 2 桁で求めた。なお、計算を続けても輪郭が変化しなくなったときを Snakes が収束したとみなし、収束した輪郭が目的の輪郭と主観的に見てほぼ一致しているときを、目的の輪郭が検出できたとみなす。また、サンプル輪郭モデルは、人間が与えた輪郭近傍で、画像エネルギーがもっとも小さくなるような輪郭を用いた。

3.4.1 ノイズの少ない画像の例

ノイズの少ない画像において輪郭がはっきりしている対象に対して、Snakes と M-Snakes とを適用することによりパラメータの変動に対する収束結果の安定性を確認する実験を行った。実験に用いたサンプル輪郭モデルを図 6(a) に示す。

はじめに、Snakes を用いた場合に目的の輪郭に収束する w_{image} の範囲を、人間がパラメータを変化させながら求めた。次に、同じ画像に対して図 6(a) をサンプル輪郭モデルとして、M-Snakes を適用し、同様に目的の輪郭に収束する w_{image} の範囲を求めた。それぞれの Snakes において、目的の輪郭に収束したパラメータの範囲を表 1 実験 3.4.1 に示す。また、輪郭の初期値を図 6(b), M-Snakes による結果および Snakes による結果を図 6(c)(d) に示す。

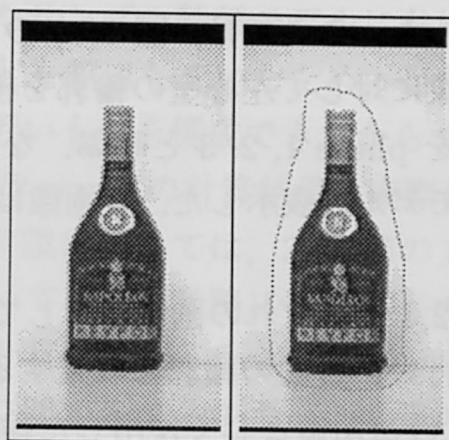
この実験では、サンプル輪郭モデルにおける画像と、輪郭を検出

		Snakes	M-Snakes	計算による範囲
実験 3.4.1		1.0 — 3.0	0.30 — 20.0	—
実験 3.4.2	phase 1	2.5 — 2.9	0.040 — 2.7	—
	phase 2	1.6	0.20 — 20.0	1.5 — 2.9
	phase 3	2.5	0.030 — 10.0	1.5 — 3.9
実験 3.4.3	phase 1	—	15 — 30	—
	phase 2	—	1.0 — 60	38.9 — 41.2

表 1: パラメータ w_{image} の範囲

した画像とが一致しているため，M-Snakes のエネルギー関数は収束時に最小性が保証されている．このような場合，表 1 からわかるように，Snakes において目的の輪郭に収束するパラメータの範囲に比べ，M-Snakes の場合 20 倍近い範囲において安定して輪郭の検出を行うことができた．エネルギー関数が単調であれば M-Snakes は，パラメータに無関係にサンプル輪郭モデルと同じ輪郭に収束するはずである．実験で，パラメータによっては目的の輪郭に収束しないのは，与えた輪郭の初期値とサンプル輪郭モデルにおける輪郭との間にエネルギー関数の極小値がある，あるいはサンプル輪郭モデルにおける輪郭においてエネルギー関数が極小値を持たないからである．

本実験では対象物の傾きがモデルと同じであるが，対象物が重心を中心とするような回転をする場合，現在の方法では対処できない．これを解決するためには，回転に依存しない対応づけを行う必要がある．



Sample
(a) Contour
Model

(b) Initial
Contour



Result of
(c) M-Snakes($w_{image} = 20.0$)

Result of
(d) Original Snakes($w_{image} = 3.0$)

図 6: ノイズの少ない画像

Fig 6: Image with less noise

3.4.2 ノイズの多い動画像の例

M-Snakes の動画像における有効性を示すため、ノイズの多い動画像に対する実験を行った。ここでは、図 7(a) の 3 枚の時系列画像である MRI 断層像画像に対して左心室の輪郭を検出した。ここでは、これらの 3 枚の画像を phase 1, 2, 3 と呼ぶ。なお、図 7(a) phase 1 には、サンプル輪郭モデルを表示した。各画像は 1/30 秒間隔で撮られている。

はじめに phase 1, 2, 3 それぞれの画像に対して、従来の Snakes を用いて輪郭の検出を行い、目的の輪郭に収束する w_{image} の範囲を求めた。なお、Snakes の初期値として使用した輪郭を図 7(b) に示す。次に、時相 1 の画像における輪郭をサンプル輪郭モデルとし、M-Snakes を phase 1, 2, 3 それぞれの画像に適用し、目的の輪郭に収束する w_{image} の範囲を同様に求めた。同時に 3.3.3 節の計算法によって phase 1, 2 の画像および輪郭を用いた場合、phase 2, 3 の画像および輪郭を用いた場合のパラメータの範囲を求めた。それぞれの Snakes において、目的の輪郭に収束する w_{image} の範囲および計算によって求めたパラメータの範囲を表 1 実験 3.4.2 に示す。M-Snakes を用いたときの、phase 1, 2, 3 における結果を、図 7(c) に示す。 w_{image} としては、計算によって得られた値の中間値を用いた。また、従来の Snakes による検出結果を図 7(d) に示す。

この実験において、phase 1 に関しては前節の実験と同様にサンプル輪郭モデルにおける画像と同じ画像に対してサンプル輪郭モデルを利用した Snakes を適用したことになるが、同様に大きな範囲の w_{image} において輪郭の検出を行うことができた。また、phase 2 及び phase 3 の画像では、従来の Snakes ではノイズの影響で目的の輪郭に収束する前にローカルミニマに収束するため非常に狭い範囲でしか良い結果が得られなかったが、M-Snakes においては、かなり広

い範囲の w_{image} において目的の輪郭に収束することが確認できた。これは、対象画像がサンプル輪郭モデルにおける画像と画像濃度が類似しており、またサンプル輪郭モデルにおける輪郭と目的の輪郭が大きさ、形状なども類似していたため、目的とする輪郭でエネルギー関数が最小に近いものを構成できたからである。

3.3.3節によるパラメータの計算結果は実際に収束する範囲よりも狭くなっている。下限に関しては、3.3.3節の方法では、計算に大局的な性質のみを用いており、経路によらず必ず目的の輪郭に収束するという最悪値を計算しているからであると考えられる。また、上限に関しては形状に関するエネルギーの計算のための輪郭を仮定しているが、実際には形状に関するエネルギーがより大きくなるような形状になることがあるためであると考えられる。

3.4.3 複雑な形状の例

人体のように変形する対象物を検出する場合、Snakes では細長くなる部分や複雑な凹凸の部分とうまく検出できないという問題があった。M-Snakes は、モデル輪郭の形状に近い形状を検出する性質があるので、このような部分に対しても有効であると考えられる。ここでは図 8(a) の二枚の歩行する人物の時系列画像から歩行者を検出した。なお、図 8(a) 左にはサンプル輪郭モデルも示した。

この画像に対して、Snakes を適用した場合、目的とする輪郭に非常に近い輪郭を初期値としても、目的の輪郭の検出は不可能であった。これは対象の凹部において、Snakes が内側に入り込まないことや、細長い形状において長軸側の輪郭がうまく検出されず内側に輪郭が入り込んでしまうことが原因である。一方 M-Snakes でもある程度目的の輪郭に近いところに初期値をおかなければならないが、それぞれの時相において表 1 実験 3.4.3 のような範囲において目的の輪郭

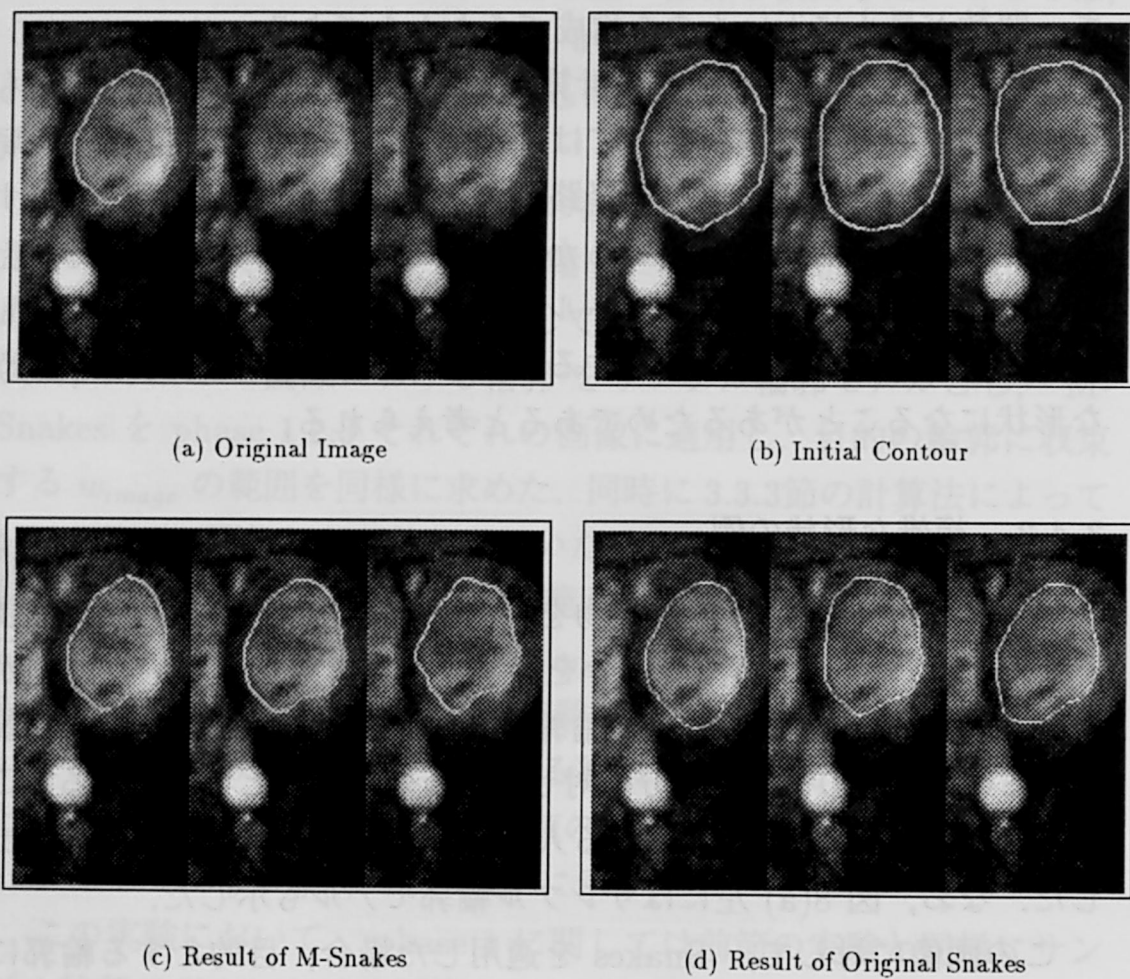


図 7: ノイズの多い時系列画像

Fig 7: Sequential Images with noise

にかなり近い輪郭を検出することができた。ただし、一部凹部では背景のエッジに捕らえられてしまう部分もあり正確に目的の輪郭を検出することはできていない。これは目的の輪郭の近くに、目的の輪郭と同様のエッジが存在するため、そちらを検出するからである。なお、対象物の輪郭の形状がかなり複雑であり二つの画像における輪郭の形状の差がやや大きいことから、式 (10) における E_{int}^0, E_{image}^0 に対して、輪郭方向にその値に対してスムージングを行った結果をそれぞれ E_{int}^0, E_{image}^0 として用いた。それぞれの時相における Snakes の初期値を図 8(b)、検出結果を図 8(c)、Snakes による結果を図 8(d) に示す。また 3.3.3 節の計算法によって求めたパラメータの範囲も表 1 実験 3.4.3 に示した。これによると、実際には計算結果に比して非常に広い範囲のパラメータで目的の輪郭に収束しており、この計算法によるパラメータによって目的の輪郭の検出を行うことができることが認められる。計算法によるパラメータの範囲が実際に収束する範囲よりも狭い理由としては、前節と同様、収束経路によるものや、計算では輪郭として単純な形状を仮定しているが、実際にはかなり複雑な形状であるため形状に関するエネルギーが仮定よりも大きいためであると考えられる。

3.5 むすび

本章では、Snakes を正則化理論の枠組からとらえなおすことによって、サンプル輪郭モデルを利用した Snakes — M-Snakes — を提案した。これは、サンプル輪郭モデルという、一つの画像における輪郭を入力することによって、その画像と同様の性質を持つ画像に対して安定した輪郭の検出を行うことができる Snakes である。従来の Snakes では、安定化汎関数として連続解を仮定した Tikhonov 型のものを用いていたが、M-Snakes では、安定化汎関数はサンプル

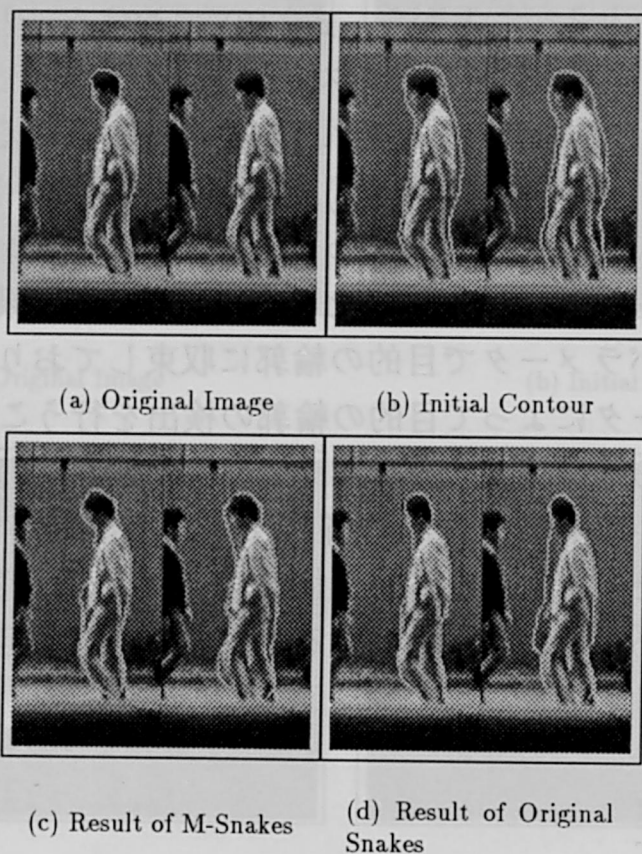


図 8: 複雑な形状の対象

Fig 8: Rough Shaped Object

輪郭モデルとして与えられる輪郭形状との差として与えられる。このため、サンプル輪郭モデルに区分的に連続な輪郭を与えることにより、区分的に連続な輪郭を解として得ることができる。また、従来の Snakes では、目的の輪郭に収束させるためのパラメータの決定が困難であったが、サンプル輪郭モデルを利用した Snakes では、収束結果に対する安定性が増加し、実際の画像に適用する場合に扱いやすいものとなっている。この結果、1) Snakes の特長であるノイズの影響を受けにくい輪郭検出の性質を残したまま、2) パラメータの変動に対して安定性が増し、3) 区分的に連続な輪郭等の複雑な形状の対象物も検出できる、輪郭検出処理を構成することができた。また、動画像などの同様の性質を持つ複数の画像に対してサンプル輪郭モデルを利用した Snakes を適用する場合、2枚の画像における輪郭を用いることによってパラメータを計算する方法を提案した。この方法によって計算されるパラメータの範囲は実際に収束するパラメータの範囲よりも狭いが、パラメータをこの範囲内に決めることによって目的の輪郭をより確実に検出できるため非常に有効な方法であると結論できる。

4 章

不連続正則化手法を用いた並行移動物体のオプティカルフロー推定

4.1 まえがき

従来，正則化手法が応用されてきた問題では，連続解を前提とした Tikhonov 型の安定化汎関数が用いられてきた．しかしながら，実際の問題では区分的に滑らかな解などの不連続解を，解の中に含む問題も多く，そのような問題に対しては良い解を与えないという問題があった．また，正則化手法では，関数の最小化問題を解くことによって最適な解を得るが，数値計算によって解を求める場合，解の初期値が必要になる．数値計算では，現実的な時間で最適解を求めることは困難であるので，一般的には局所最適解を求めることになるが，この場合，初期値によって解が変動することになる．従って，実際の問題に対して正則化手法を適用する場合，何らかの方法で良い初期値を与える手法が同時に必要になる．

前章では，区分的に滑らかな解に対処する手法として，不連続な解のモデルを与え，モデルとの誤差関数に対して従来の正則化手法を適用する手法を提案した．本章では，不連続な分布を持つオプティカルフローの推定問題を考え，この問題に対し，離散値を取る安定

化汎関数を導入することにより，問題の物理的な制約に合う解を与える正則化手法を提案する．さらに，問題の制約から正則化手法とは独立な手法を用いて良い初期値を与える手法を検討し，それを用いることにより，妥当な解が得られることを示す．

運動している対象物の，画像上の見かけの運動を表すオプティカルフローは，計算機による画像認識のために有用な情報である．この問題は画像情報のみを用いたのでは解が一意に定まらない不良設定問題であるため，画像以外の様々な制約や知識を導入することが行われてきた．

エッジを用いる方法 [24][25] [26][27] では，はじめに運動している対象物のエッジを検出し，得られたエッジの対応をとることによってオプティカルフローを求めている．しかしながら，これらの方法は，エッジの検出精度によって，結果が大きく影響を受けるため，安定したエッジ検出を行うことができる環境でしか利用することができない．

また，領域を用いる方法には Markov Random Field を利用したもの [28][29][30][31] やピラミッド法を用いたもの [32] などがあるが，それぞれ計算量が膨大となる，あるいは必ずしも正確な領域のマッチングが得られる保証がないなどの問題がある．

クラスタリングを用いる方法としては，FOE を用いる方法 [33]，Schnörr の方法 [34]，Agarwal の方法 [35] などがある．FOE を用いる方法では各移動物体のエッジを検出し，同じ並行運動をしている一組のエッジは同じ FOE をもつという性質を利用して，同じ物体に属するエッジをセグメンテーションにより求め，オプティカルフローを求めている．この方法は，はじめにエッジの検出を行い，その結果を用いるため，エッジを基本にした手法と同様の問題が生じる．また，Schnörr の方法は，同じ物体に属すると確実に判断できる，十分小さ

な領域を、輝度一定の条件を満たさなくなるまで拡大するという操作により、画像中の全画素を、移動物体ごとに領域分割する方法であるが、領域の拡大操作において、ノイズの影響を受けやすいという問題がある。さらに、Agarwalの方法では他の方法によりオプティカルフローの初期値を求め、これをクラスタリングした結果に対して近傍領域のマッチングをとることによって、オプティカルフローの反復改良をおこなっているが、反復改良の方法が領域を基本としたテンプレートマッチングであり、画素単位の移動しか検出できないという問題がある。

正則化手法を用いた方法は、理論的根拠が明確であるため、能力が明らかであるという特長がある。正則化手法を用いる方法としてはHornらによる方法[6]がある。この方法は、アルゴリズムの能力が明らかであり、画像のノイズ分布等の性質が明らかであれば、非常に有効な方法であるといえる。しかしながら、この方法ではオプティカルフローが滑らかに変化するという仮定を用いているため、オプティカルフローが不連続に変化するような問題に対しては良い解を与えない。つまり、この方法では、オプティカルフローが不連続に変化する部分において、その変化を滑らかに補間しようとするため、不連続に変化する領域の周囲に誤差が大きく伝播する。従って、この手法は適用できる問題の範囲が非常に狭いという問題があるといえる。

このような問題に対して2章で述べたように、不連続な解分布を扱う、不連続正則化手法がいくつか提案されている。これらの手法では、解の不連続性に関して、不連続部分の画像上の位置、不連続部分の統計的な性質等の情報が事前に必要であるが、これらの情報が必ずしも、手法を適用しようとしている問題から事前に得られるとは限らない。従って、手法を適用しようとしている問題を検討し、事前に得られる情報がうまく利用できる手法を使う必要がある。

ここでは、解空間の中に不連続部分を含む解が存在する関数を考え、これを安定化汎関数として用いる正則化手法により、不連続部分を含むオプティカルフローを推定する手法を提案する。特に、画像中の移動物体は並行移動のみを行っているという制約を表現する安定化汎関数として、離散的な値のみを取るような関数を考え、これを用いる手法を提案し、実験を行った。また、正則化手法には初期値の決定が重要な要素となるが、ここでは勾配法における拘束線の分布による初期値計算法を用いた。特に従来あまり検討されなかった誤差について詳しく検討を行うことによって解ける問題の範囲を明らかにした。

4.2 不連続部を含むオプティカルフローの推定

4.2.1 オプティカルフロー

オプティカルフローとは、物体の画像上の見かけの速度ベクトルのことである。画像の輝度を $I(x, y)$ とし、各画素に対応するオプティカルフローを (u, v) とすると、以下の関数を最小化することによってオプティカルフローを求めることが出来る。

$$\int \int ((I_x u + I_y v + I_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)) dx dy$$

移動物体を対象とする処理においては、背景の前を平行あるいは回転運動しているものを扱うことが多いが、Tikhonov 型の安定化汎関数を用いた処理では、このような場合に対処できない。

4.2.2 極小値が全て 0 になる関数を安定化汎関数として用いた正則化

Tikhonov 型の安定化汎関数を用いると、解が不連続になる部分において正しい解が得られない。オプティカルフローでいえば、移動物

体と背景画像の境界あるいは、移動物体と移動物体の境界において、正しい解が得られない。これは、解が Tikhonov 型の関数で表現される空間の中に存在しないからである。

ここでは、この問題に対処するため、不連続部分を含む解を解空間の中に含む関数を考え、これを安定化汎関数として用いることを考える。ここでは次のような制約を表現する関数を用いる。すなわち、画像中の移動物体は高々数個であり、それぞれの移動物体の運動は、いくつかの既知の運動のいずれかに属する。

このような制約を表現する安定化汎関数として以下のようなものを用いることを提案する。

$$\prod_{i=1}^h g(f_i(w)) \quad (16)$$

w はオプティカルフロー (u, v) を表し、 $f_i(w)$ ($i = 1, 2, \dots, h$) はオプティカルフロー w が既知の運動 i に属するとき、値が 0 になる関数である。また h は運動の数を示す。 f_i としては、画像上速度 W で並行運動しているものに対応する、

$$f(w) = (w - W)^2 \quad (17)$$

などが考えられる。

関数 $g(x)$ の満たすべき条件は次のとおりである。

1. $x = 0$ で唯一つの極小値をもち、その値は 0 である。
2. 左右対象であり、十分大きい x に対して、 $g'(x) \approx 0$ である。

各画素に対応するオプティカルフロー w は f_1, f_2, \dots, f_h のどれかの値を 0 にする値を取るが、このような関数を安定化汎関数として用いると、オプティカルフロー分布の空間的な情報を用いないこ

となる．このため，従来の手法では分布の不連続部分で生じていた問題を回避することができる．

式(16)を安定化汎関数として用いるオプティカルフローの計算は，以下に示す式の最小化によって求められる．

$$R = \iint \left\{ (I_x u + I_y v + I_t)^2 + \lambda \prod_{i=1}^h g(f_i(\mathbf{w})) \right\} dx dy \quad (18)$$

ここでは先の条件を満たす関数として次の関数 $g(x)$ を用いる．

$$g(x) = \begin{cases} ax^2 & 0 \leq |x| < c \\ -\frac{p}{(x-q)^2} + 1 & c \leq |x| \end{cases} \quad (19)$$

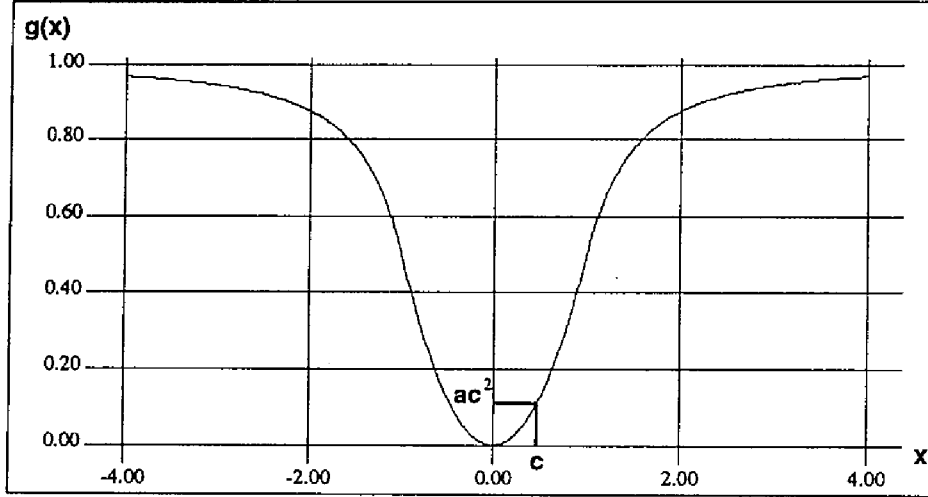
ただし， a, c, q は， $a, c, q > 0, q < c$ を満たす．この関数 $g(x)$ が x に関して滑らかな連続関数であるためには， $p = \frac{(1-ac^2)^3}{(ac)^2}, q = \frac{2ac^2-1}{ac}$ である必要があり，このとき $g(x)$ は以下のように表される．(図 9 参照)

$$g(x) = \begin{cases} ax^2 & 0 \leq |x| < c \\ -\frac{(1-ac^2)^3}{(ac)^2 \left(x - \frac{2ac^2-1}{ac}\right)^2} + 1 & c \leq |x| \end{cases} \quad (20)$$

4.2.3 オプティカルフロー計算

式(18)を最小にする (u, v) 及び f_i に含まれるパラメータを反復計算により求め，オプティカルフローの計算を行う．この計算における \mathbf{w} の初期値としては，他の手法で得られるものを用いる．

$\mathbf{I}_w = (I_x u + I_y v + I_t) \begin{pmatrix} I_x \\ I_y \end{pmatrix}$ ， $G_i = \prod_{\substack{j=1 \\ j \neq i}}^h g(f_j(\mathbf{w}))$ とすると，式(18)を最小にする \mathbf{w} は以下の式を満たす．

図 9: 関数 $g(x)$ Fig 9: Function $g(x)$

$$\frac{\partial R}{\partial \mathbf{w}} = \iint \left\{ 2\mathbf{I}_w + \lambda \sum_{i=1}^h g'(f_i(\mathbf{w})) f'_i(\mathbf{w}) G_i \right\} dx dy = 0 \quad (21)$$

同様に f_i に含まれるパラメータを \mathbf{r} とすると, 式 (18) を最小にする \mathbf{r} は以下の式を満たす.

$$\frac{\partial R}{\partial \mathbf{r}} = \iint \left\{ \lambda \sum_{i=1}^h g'(f_i(\mathbf{w})) \frac{\partial f_i}{\partial \mathbf{r}} G_i \right\} dx dy = 0 \quad (22)$$

従って, \mathbf{w} 及び \mathbf{r} は, 適当な初期値 $\mathbf{w}^{[1]}$ 及び $\mathbf{r}^{[1]}$ を用いて, 以下のような反復計算を行うことにより求めることができる.

$$\mathbf{w}^{[k+1]} = \mathbf{w}^{[k]} - \frac{\partial R}{\partial \mathbf{w}} \quad (23)$$

$$\mathbf{r}^{[k+1]} = \mathbf{r}^{[k]} - \frac{\partial R}{\partial \mathbf{r}} \quad (24)$$

以上の計算によって，式 (18) を最小にする (u, v) 及び，関数 f_i に含まれるパラメータを決定することができる．

4.3 勾配法を用いた並行移動領域のオプティカルフローの初期値推定

4.3.1 勾配法における拘束線

前節で説明した手法には，反復計算の初期値として，画像上で運動している領域の運動パラメータの値が必要である．ここでは，輝度勾配を用いた並行移動領域のオプティカルフローの初期値推定方法について説明する．

輝度勾配を用いたオプティカルフローの推定方法は，Horn and Shunk[6] によって提案された．

時刻 t における，画像上の点 (x, y) に対応する物体上の点の輝度を $I(x, y, t)$ とすると，以下の式が成り立つ．

$$\frac{dI}{dt} = I_x u + I_y v + I_t = 0 \quad (25)$$

ここで， (u, v) は画像上の点の見かけの速度を表し，オプティカルフローと呼ぶ．

ところで，画像上の全ての点で上式の I_x, I_y, I_t は求まるが，式は1つで未知変数が2つあるため， (u, v) は一意には定まらず，ある直線上にのみ拘束される．この直線を拘束線 (constraint line) と呼ぶ．

画像上に，同一方向に並行移動をしている領域がある場合，この領域内の点のオプティカルフローは全て同じになる．従って，これらの点に対応する拘束線は全て1点で交わる．このような条件を満たす場合，拘束線同士の交点の分布を見ることによって，並行移動領域のオプティカルフローを決定することができる．しかしながら，

実際には画像のノイズ，計算誤差等により，交点は必ずしも 1 点で交わるとは限らない．また，領域の境界部分は，輝度勾配などが正しく計算できないため，式 (25) を満たさない．

ここでは，上記の方法を用いるときに，差分による計算誤差及び，量子化誤差によって拘束線に生じる変位を計算し，画像から得られるオプティカルフローの精度を求める．さらに，複数の並行移動領域が存在するときに，それらのオプティカルフローが安定して求められるための条件を導く．

4.3.2 差分による輝度勾配の計算誤差

画像の空間周波数 ある物理空間の平面への写像を $E(x, y, t)$ とする．ただし， $0 \leq x < 1, 0 \leq y < 1$ とし， t は時間を表す．このときこの写像をサンプリングした画像 $I(x, y, t)$ は一般に以下のように表される．

$$I(x, y, t) = E(x, y, t) * w(x, y, t) \quad (26)$$

ここで， $*$ はたたみ込み演算であり， $w(x, y, t)$ はサンプリング関数である．信号理論などでは，サンプリング関数としてインパルス関数を用いるシャノン型のサンプリングを考えることが多いが，CCD などの光電変換素子を空間的に並べた撮像素子においては，このサンプリング関数は空間のある範囲で一定値を取り，それ以外の所では 0 となるような関数で近似される．つまり，ある光電変換素子の出力は，その素子上に入射している光の明るさの平均になっていると考えられる．このようなサンプリングを Cahn サンプリングと呼ぶ．Cahn サンプリングでは，シャノン型のサンプリングと違い，入力に対して低域通過フィルターをかけたのと同じ効果がある．従って，シャノン型のサンプリングではもともとのデータに含まれる最高

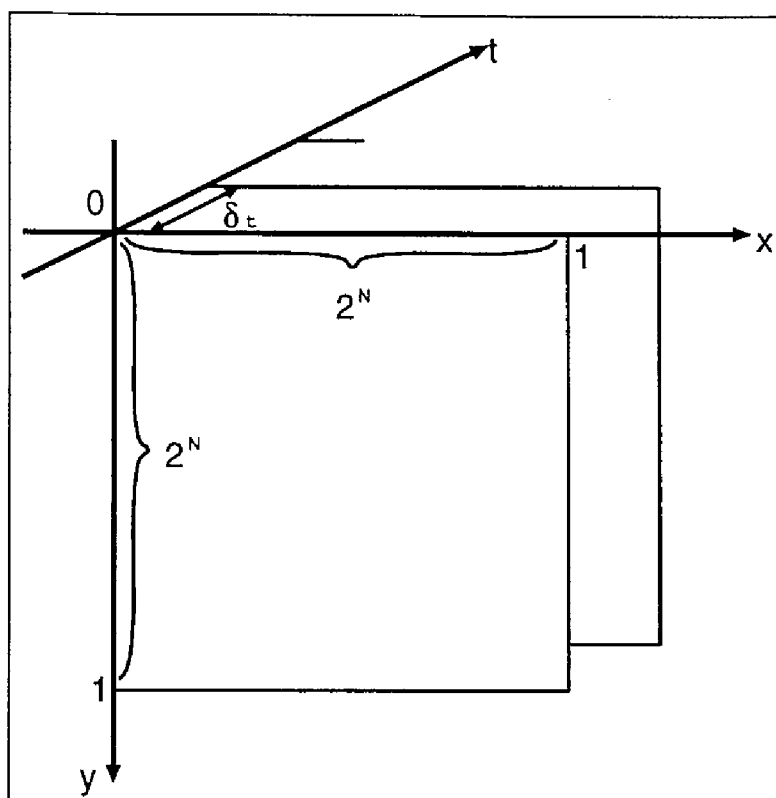


図 10: 座標系

Fig 10: Coordinates

空間周波数は未知であるのに対して, Cahn サンプリングでは, データに含まれる最高空間周波数はサンプリングで復元できる最高空間周波数と一致する.

いま, 画像 $I(x, y, t)$ は空間的に $2^N \times 2^N$, 時間的に δ_t でサンプリングされているとする. するとこの画像に含まれる最高空間周波数は, $f = 2\pi 2^{N-1}$ となる. また, 時間方向の周波数は, $f = \frac{2\pi}{\delta_t}$ となる. (図 10 参照)

輝度勾配の推定誤差 通常勾配法によってオプティカルフローを求める場合、各画素の輝度勾配を差分法によって求める。ここでは、輝度勾配を中心差分によって求める場合について考える。

中心差分では、関数 $g(x)$ における、点 x_i の勾配をその前後の標本点の値から、以下のように求める。

$$\left. \frac{dg}{dx} \right|_{x=x_i} = \frac{g(x_{i+1}) - g(x_{i-1}))}{2\Delta x}$$

ただし、 $x_{i+1} - x_i = \Delta x$ とする。ところで、最高周波数成分 $2\pi n$ を含む信号は

$$g(x) = \sum_{j=1}^n A_j \cos(2\pi j x + \theta_j)$$

と表される。 $g(x)$ の勾配 $g'(x)$ は以下のようにになる。

$$g'(x) = \sum_{j=1}^n -2\pi j A_j \sin(2\pi j x + \theta_j)$$

ただし、 $\max_x |g(x)| = 1$ とする。

ここで、中心差分による勾配の計算と、 $g'(x)$ との誤差を計算すると以下のようにになる。

$$\begin{aligned} e(x) &= g'(x) - \left. \frac{dg}{dx} \right|_{x=x} \\ &= - \sum_{j=1}^n 2\pi j A_j \sin(2\pi j x + \theta_j) - \frac{g(x + \Delta x) - g(x - \Delta x)}{2\Delta x} \\ &= - \sum_{j=1}^n 2\pi j A_j \sin(2\pi j x + \theta_j) \\ &\quad - \frac{\sum_{j=1}^n A_j \cos(2\pi j(x + \Delta x) + \theta_j) - \sum_{j=1}^n A_j \cos(2\pi j(x - \Delta x) + \theta_j)}{2\Delta x} \\ &= - \sum_{j=1}^n 2\pi j A_j \sin(2\pi j x + \theta_j) \end{aligned} \tag{27}$$

ρ	1/2	1/4	1/8
ε_d	0.57080	0.07829	0.01002

表 2: 中心差分の誤差

$$\begin{aligned}
& + \sum_{j=1}^n A_j \frac{\sin(2\pi j \Delta x)}{\Delta x} \sin(2\pi j x + \theta_j) \\
& = - \sum_{j=1}^n A_j \left(2\pi j - \left(\frac{\sin(2\pi j \Delta x)}{\Delta x} \right) \right) \sin(2\pi j x + \theta_j)
\end{aligned}$$

$\sum A_j = 1$ であるから, 上式を最大にするのは, $\sin(2\pi j x + \theta_j) = 1$ であつ, $A_j = 0 (j = 0, \dots, n-1), A_n = 1$ のときである. 従つて e の最大値は以下のようなになる.

$$\max e = 2\pi n - \frac{\sin(2\pi n \Delta x)}{\Delta x} \quad (28)$$

サンプリングによって復元できる最高空間周波数を $2\pi N$ とすると, 通常 Δx は,

$$\Delta x = \frac{1}{2N} \quad (29)$$

とあらわされる. 従つて, 信号に含まれる最高空間周波数成分 $2\pi n$ とサンプリングによって復元できる最高周波数成分 $2\pi N$ との比 $\rho (= n/N)$ によって生じる, 実際の勾配と中心差分による勾配との差 $\varepsilon_d = |\max e - \Delta x|$ を式 (28) 及び式 (29) により計算すると, 表 2 のようになる.

この表から通常の Cahn サンプリングした画像では, 中心差分による勾配の計算では最大 57% 程度の誤差が生じることがわかる. 逆に, 輝度勾配に必要な精度が定まれば, 原画像に対して低域通過フィルターをかけることによりこれを実現することができる.

ここまでは、空間方向への輝度勾配について考えたが、時間軸方向への輝度勾配についても同じように中心差分による誤差を計算することができる。時間軸方向へのサンプリング間隔は δ_t であるから、復元できる最高周波数成分は $f_t = \frac{1}{2\delta_t}$ である。

ところで、空間方向への周波数成分と時間軸方向への周波数成分は、同じ信号を違う軸で観測しているだけであるため、互いに関連がある。空間方向に最大 f_s の周波数成分を持つ信号が速度 u で移動しているとすると、この信号は時間軸方向には最高周波数成分 uf_s を持つ信号として観測される。時間軸方向の輝度勾配を計算するためには、 $uf_s < f_t$ を満たさなければならない。また、 uf_s と f_t の比を $\rho(= uf_s/f_t)$ とすると、時間軸方向への輝度勾配の誤差も空間方向への輝度勾配の誤差 ε_d と同じになる。

4.3.3 量子化誤差

標本化された信号には量子化による量子化誤差が含まれる。輝度の最大値を 1 とすると、 2^k 段階で一様量子化した場合の、中心差分による勾配の最大誤差 ε_q は $1/2^{k-1}$ となる。この誤差は信号強度とは関係なく、加法的に加えられる。

256 階調で量子化した場合、 ε_q は $2^{-7} = 0.00781$ 程度になる。

4.3.4 輝度勾配の誤差と拘束線の変位

ここで、輝度勾配の誤差によって拘束線がどの程度変位するかを考える。画像上のある点に対応する拘束線を $I_x u + I_y v + I_t = 0$ とし、この点のオプティカルフローを (p, q) とすると、拘束線とこの点との距離は、 $I_x, I_y \neq 0$ のとき以下のように表される。

$$d = \frac{I_x p + I_y q + I_t}{\sqrt{I_x^2 + I_y^2}}$$

輝度勾配が誤差を含まないとき、上式は 0 となる．ここで、 I_x が差分による誤差 ε_{dx} と量子化による誤差 ε_{qx} を持ち、 $(1 + \varepsilon_{dx})I_x + \varepsilon_{qx}$ であるとする．同様に I_y も $(1 + \varepsilon_{dy})I_y + \varepsilon_{qy}$ であるとする．このとき、拘束線と点 (p, q) の距離は、以下のようになる．

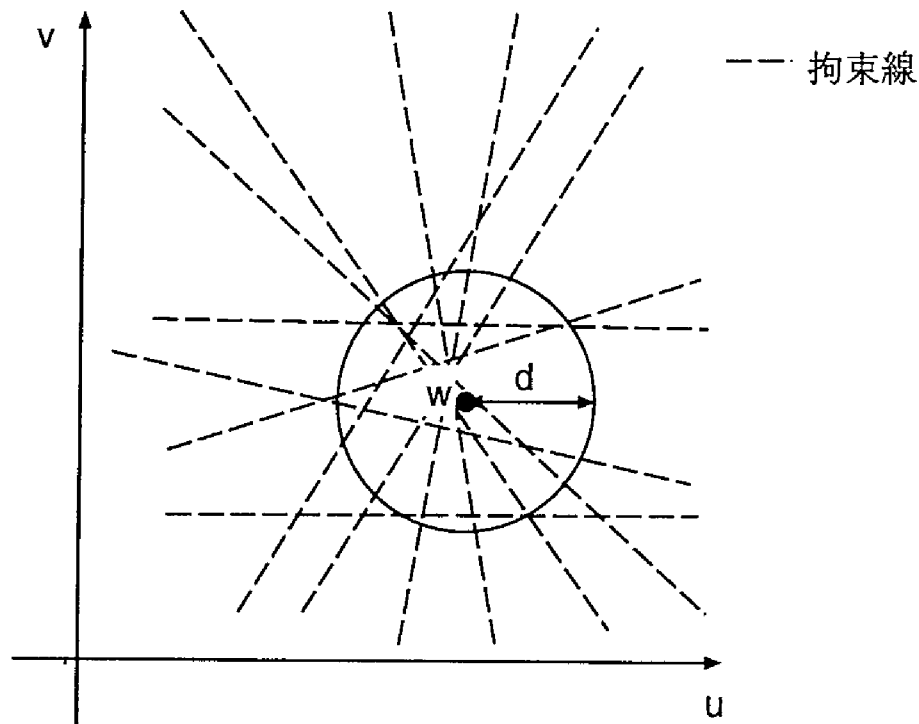
$$d = \frac{(\varepsilon_{dx}I_x + \varepsilon_{qx})p + (\varepsilon_{dy}I_y + \varepsilon_{qy})q}{\sqrt{((1 + \varepsilon_{dx})I_x + \varepsilon_{qx})^2 + ((1 + \varepsilon_{dy})I_y + \varepsilon_{qy})^2}} \quad (30)$$

この式より、オプティカルフローの最大値が既知であれば、拘束線の変位 d は I_x, I_y の関数として表されることがわかる．従って、ある精度でオプティカルフローを求める場合、条件に合う I_x, I_y を持つ画素のみを処理の対象とすれば良いことがわかる．また、そのような画素が少ない場合、量子化 bit 数を増やす、あるいは画像に含まれる最高空間周波数成分を落すことにより、同様の効果を得ることができる．なお、次節で投票によって求められるオプティカルフローの精度と、拘束線の変位との関係について検討する．

4.3.5 拘束線の変位を考慮した初期値計算

前節までで、画像に含まれる信号の最高空間周波数及び、量子化によって輝度勾配に生じる誤差を定量的に解析し、拘束線がどの程度変位するかを定量的に示した．ここでは、正しいオプティカルフローからの変位がわかっている拘束線を用いて、オプティカルフローを推定する方法について検討する．

速度 w で並行移動している領域の画素に対応する拘束線の中から、 w からの距離が d 以内となるものを、オプティカルフローの速度を表す UV 空間に描くと、全ての拘束線は w の d 近傍を通過する．従って、変位が d 以内である、全ての拘束線を UV 空間に描き、 UV 空間において d 近傍を通過する拘束線の数が増大となる点を探

図 11: w の d 近傍を通る拘束線Fig 11: Constraint Lines within d from w

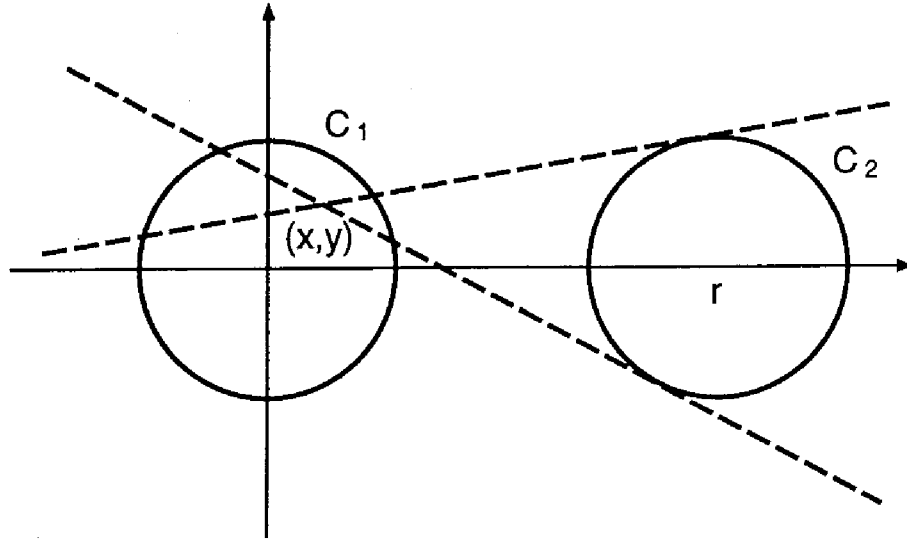
索すれば、画像上の移動領域に対応するオプティカルフローが求められると考えられる。ただし、この方法では、移動領域に対応するオプティカルフロー以外の点においても極大点が生じる可能性がある。したがって、得られる極大点はオプティカルフロー候補と呼ぶ。(図 11 参照) オプティカルフローの大きさに制約を設けない場合、解が存在する可能性のある空間は制限できなくなり、計算量が増大するので、オプティカルフローの取り得る範囲、つまり物体の見かけの最大速度は与えられているとする。

輝度勾配は画素の 4 近傍を用いて中心差分によって求めているが、輝度勾配が計算できるためには、それらの点が全て同じ並行移動領

域内の点である必要がある。逆にいえば，並行移動領域と背景との境界あるいは並行移動領域と並行移動領域との境界の画素に対応する拘束線は背景と移動領域，あるいは移動領域同士の輝度差によって決まる拘束線となる。領域の輝度に関する事前の知識がない場合，そのような領域に対応する拘束線の性質は仮定し難いので，ここではランダムな直線となると考え，一様に分布するノイズ成分と考える。この拘束線の本数は，それぞれの領域の境界の前後 3 画素分に対応するので，境界長の 3 倍となる。

上記の方法でオプティカルフローを推定する場合，拘束線の変位 d をどう決定するかが問題になる。 d を小さくすると，式 (30) を満たす画素は少なくなる。投票結果のピークを検出することによってオプティカルフローを求めることを考えると，投票数は多いほど検出結果の精度は良いと考えられ，大きな d を用いることが望ましい。一方， d を大きくした場合，各領域に対応する投票結果の分布は投票空間上に幅広く分布することになる。複数の異なる速度で移動している領域のオプティカルフローを検出することを考えると，これらの d が大きいと，これらの投票結果から，ピークを検出できるかという問題が生じる。つまり，2 種類のオプティカルフローで移動している領域が存在しているときに，これらの領域の画素に対応する拘束線が互いに影響を受けて，うまく極大点を検出できなくなる可能性がある。ここでは，2 種類のオプティカルフローの分離可能性について検討を行う。

オプティカルフロー $(0, 0)$ から半径 d の円 C_1 を通過する拘束線の中で， $(r, 0)$ を中心とする半径 d の円 C_2 を通過するものの割合を考える。拘束線の傾きはランダムであるとする， C_1 内の点 (x, y) を通過する拘束線のうち， C_2 を通過するものの割合を求め，これを積分すればこの割合が求まる。点 (x, y) を通過する拘束線のうち C_2

図 12: 円 C_1 を通過する拘束線のうち円 C_2 を通過する割合Fig 12: The Ratio of lines passing C_2 within lines passing C_1

を通過する割合は以下のように表される. (図 12 参照)

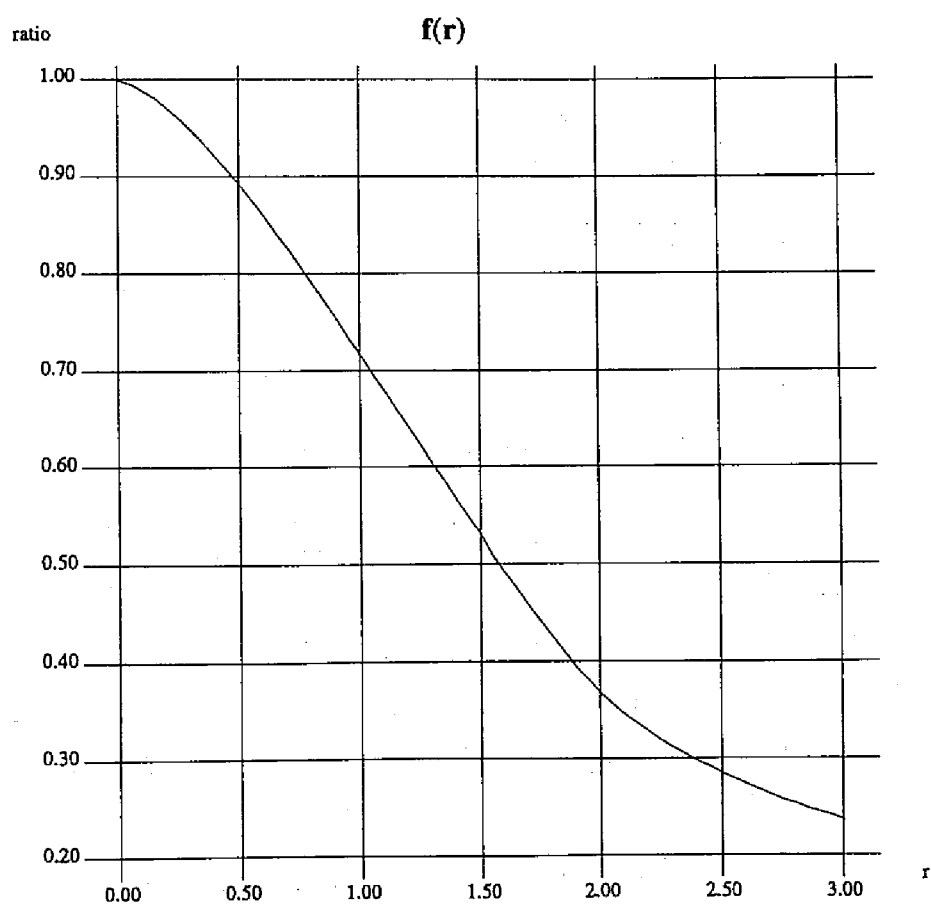
$$\phi(r, d, x, y) = \begin{cases} \frac{2}{\pi} \sin^{-1} \frac{d}{\sqrt{(r-x)^2 + y^2}} & \sqrt{(r-x)^2 + y^2} > d \\ 1 & \text{otherwise} \end{cases} \quad (31)$$

$d = 1$ のとき, C_1 を通る拘束線のうち C_2 を通るものは以下のように表される.

$$f(r) = \int_{-1}^1 \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} \phi(r, 1, x, y) dy dx \quad (32)$$

ただし, C_1, C_2 の半径が d の場合, 比率は $d f(r/d)$ になる. $f(r)$ のグラフを図 13 に示す.

画像上の二つの領域が速度 w_1, w_2 で並行移動しているとする. ただし, $|w_1 - w_2| = \alpha$ とする. その領域中境界部分を除く面積, つまり正しいオプティカルフローの d 近傍を通過する拘束線の数

図 13: $f(r)$

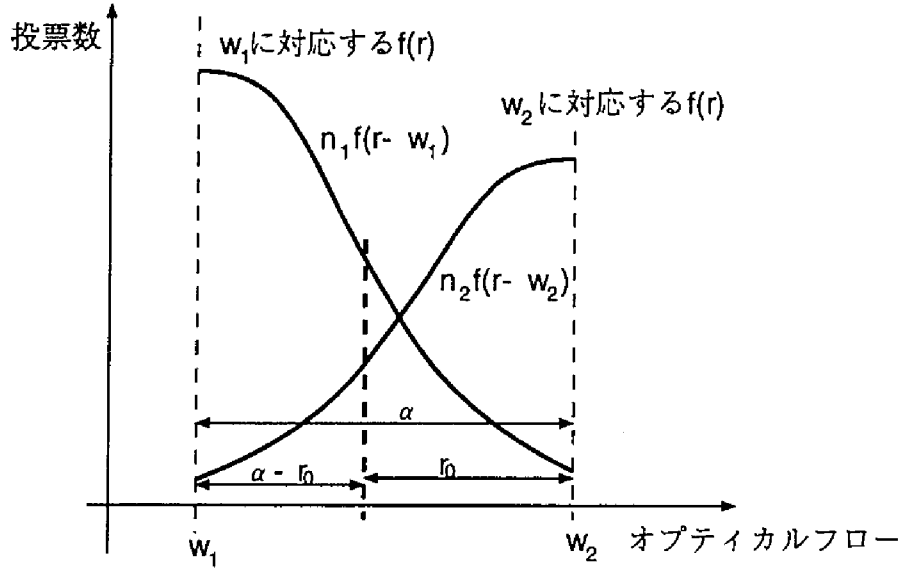


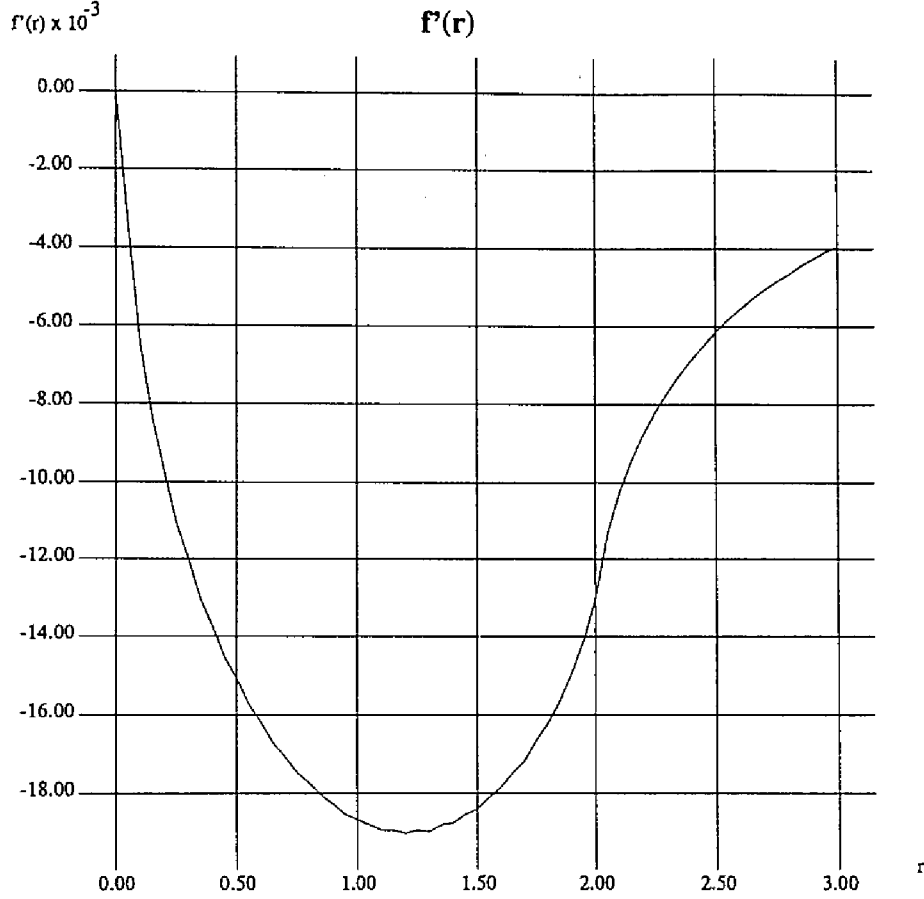
図 14: 2 種類のオプティカルフローが存在する投票空間

Fig 14: Voting Space of 2 Optical Flows

n_1, n_2 ($n_1 \geq n_2$) とする. 半径 d の円内を通過する拘束線の数 of 極大点を探すことにより, オプティカルフロー候補を求める場合, これら二つのオプティカルフローを分離するためには, これらの間に拘束線の数 of 極小点がないといけない. これは, w_1 近傍において, 対応する $f(r/d)$ の傾きの絶対値がもう一つのオプティカルフローに対応する $f(r/d)$ の傾きより大きくなければならないということである (図 14 参照). ここで $d=1$ とし, $f'(r)$ が最大になる点を r_0 とすると, これは以下のように表される.

$$-1.0 < \frac{n_1 f'(\alpha - r_0)}{n_2 f'(r_0)} < 0 \quad (33)$$

図 15 に示す $f'(r)$ のグラフによると, $r_0 = 1.2$ で, $f'(r) = -0.19$ である. このグラフより, ふたつの領域の面積の比 $m = n_1/n_2$ が既知である場合に分離可能なオプティカルフローの距離 α を $R(m)$ と

図 15: $f'(r)$

して求めることができる．このグラフを図 16 に示す．ここで， α は $d = 1$ の時の値であるので，実際の分解能は $\alpha \times d$ になる．

また，画像上に k 個の並行移動領域が存在し，それぞれの領域に対応するオプティカルフローを w_l ，その領域の面積を n_l とすると，全てのオプティカルフローの初期値が投票空間上でピークを持つためには，以下の条件を満たさなければならない．

$$n_i f'(r_0) > \sum_{j \neq i} n_j f'(\|w_j - w_i\| - r_0) \quad (34)$$

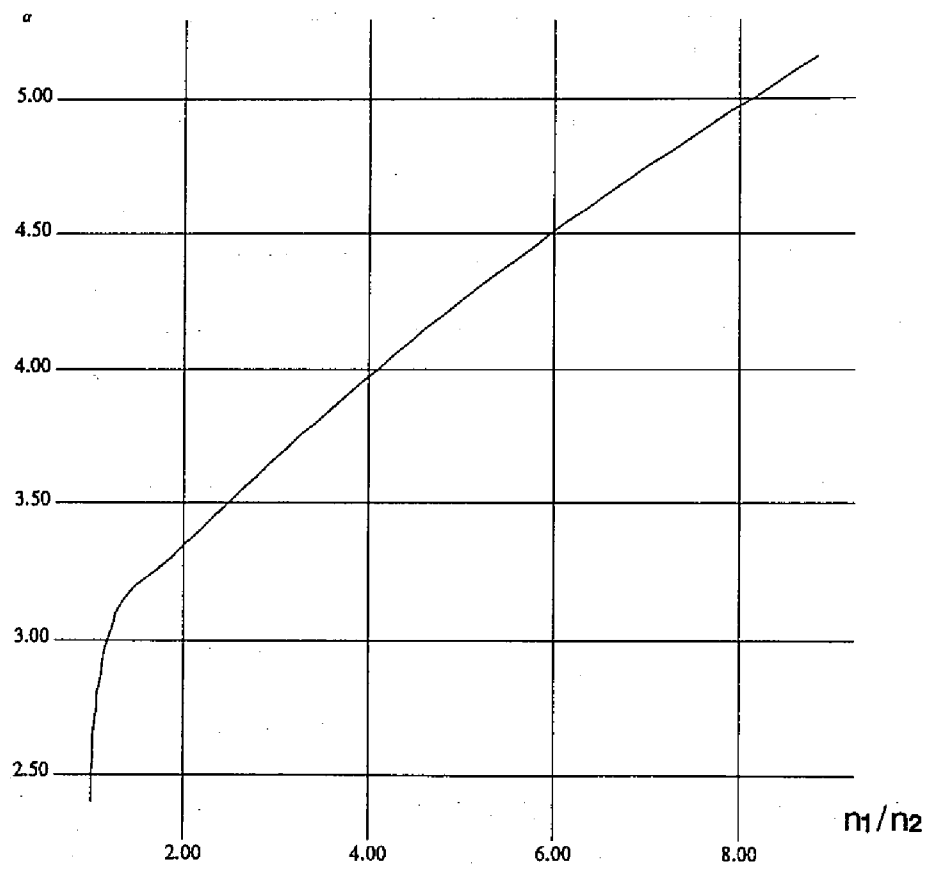


図 16: 領域間面積比が m のときのオプティカルフローの分解能 α

Fig 16: Optical Flow Resolution at Ratio of Two Regions r

最後に初期値計算の手順と条件をまとめる．

複数の並行移動している領域の，面積を n_i ，オプティカルフローを w_i とする．これらの領域のオプティカルフローの初期値を投票によって求めるためには，全ての領域の面積，オプティカルフローは式 (34) を満たさなければならない．なお，各拘束線の変位 d は，画像中の最高空間周波数と復元可能な空間周波数成分との比，および量子化 bit 数を用いて，式 (30) により求められる．

撮影環境より，領域の個数 I ，各領域の面積比の最大値 m ，オプティカルフロー間の距離の最小値 Δ_w が既知である場合は以下の手順により初期値を求めることができる．

1. 以下の式を満たす d を求める

$$R((I-1)m)d < \Delta_w \quad (35)$$

2. 式 (30) を満たす画素に対応する拘束線を用いて UV 空間に投票を行う
3. UV 空間において半径 d 内の拘束線の本数を数える (実際には，2 において幅 $2d$ の帯を投票する)
4. 半径 d 内の拘束線の本数が極大となる点を数え上げる

また，事前に m あるいは Δ_w がわからない場合， d を，式 (30) を満たす画素数が十分な数になるように決定することが考えられる．この場合，検出できるオプティカルフローの初期値は $R((I-1)m)d < \Delta_w$ という範囲に制約されることになる．

4.4 実験結果

4.4.1 勾配法による初期値推定

ここでは、4.3節の手法で初期値を推定した結果を示す．実験には図 17 のような三つの並行移動をしている領域を持つランダムドットパターンを用いた．なお、画像は 256 階調の濃淡画像であり、画像に含まれる最高空間周波数成分と復元できる最高空間周波数との比 ρ は $1/8$ である．このとき、 $\varepsilon_d = 0.01002$ であり、 $\varepsilon_q = 0.00781$ である．

データより、領域の個数 I は静止している領域を含めて 4、領域の面積比の最大値 m は 3.0 である．また、オプティカルフロー間の距離の最小値 Δ_w は 0.5 である．従って、式 (35) を満たすような d を求めると、 $d = 0.091$ となる．

式 (30) を満たす画素のみを用いて、拘束線の投票を行ったときの、投票空間の分布を図 18 に示す．なお、図は中央が原点であり、横軸は x 方向のフローベクトル軸、縦軸は y 方向のフローベクトル軸を示しており、それぞれ ± 1 の範囲を表している．また、各拘束線は幅 d の帯として投票している．このとき、拘束線の数の極大値を求めると、表 3 のようになり、誤差は最大 0.14 程度であった．複数の移動領域が存在しない場合、誤差は d 程度になるはずであるが、極大点の近くに、投票数の多い他の極大点が存在する場合、その影響で誤差が増大する．実験結果によると、この誤差は d の 2 倍以内には収まっており、初期値として妥当なものが得られていると考えられる．

また、拘束線の変位 d の影響を調べるため、 $d = 0.009$ および $d = 0.9$ のときの投票空間の分布を図 19、図 20 に示す．この図からもわかるように $d = 0.9$ の時は、投票のピークが $(0.0, 0.0)$ にしか現れなかったため、オプティカルフローの初期値を求めることができ

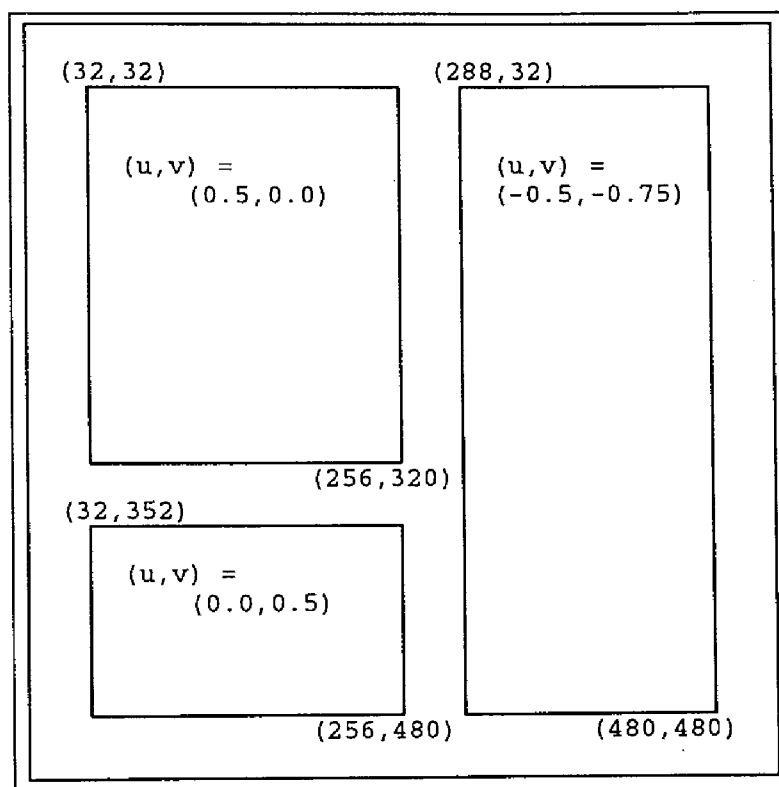


図 17: 原画像のオプティカルフロー

Fig 17: Optical Flow of Original Image

極大点 (u, v)	拘束線の数
$(0.008, -0.008)$	48000
$(-0.429, -0.609)$	41824
$(0.570, -0.043)$	36640
$(-0.117, 0.527)$	29616

表 3: 拘束線の数 of 極大点 ($d = 0.091$)

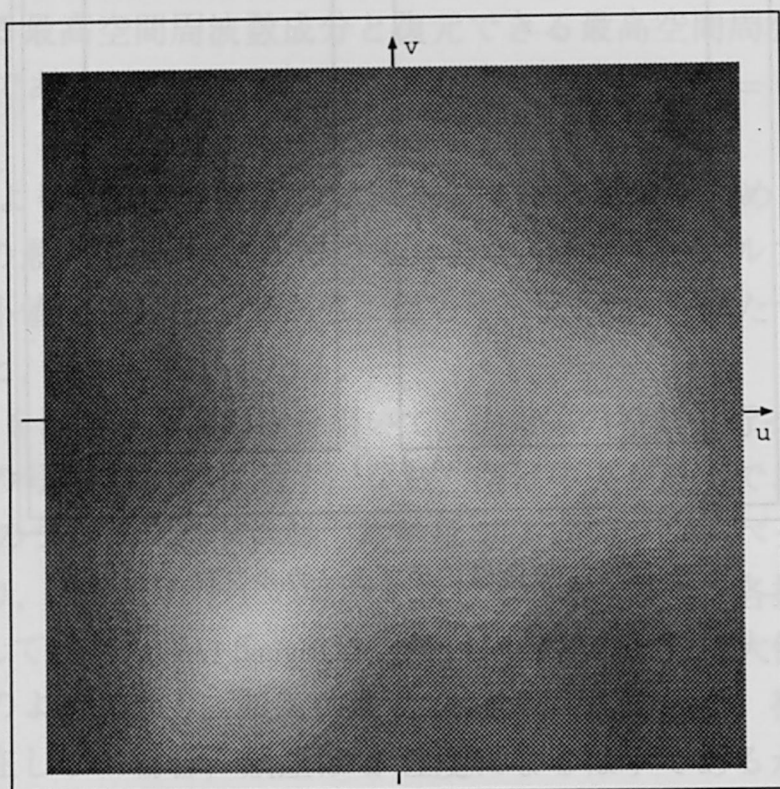
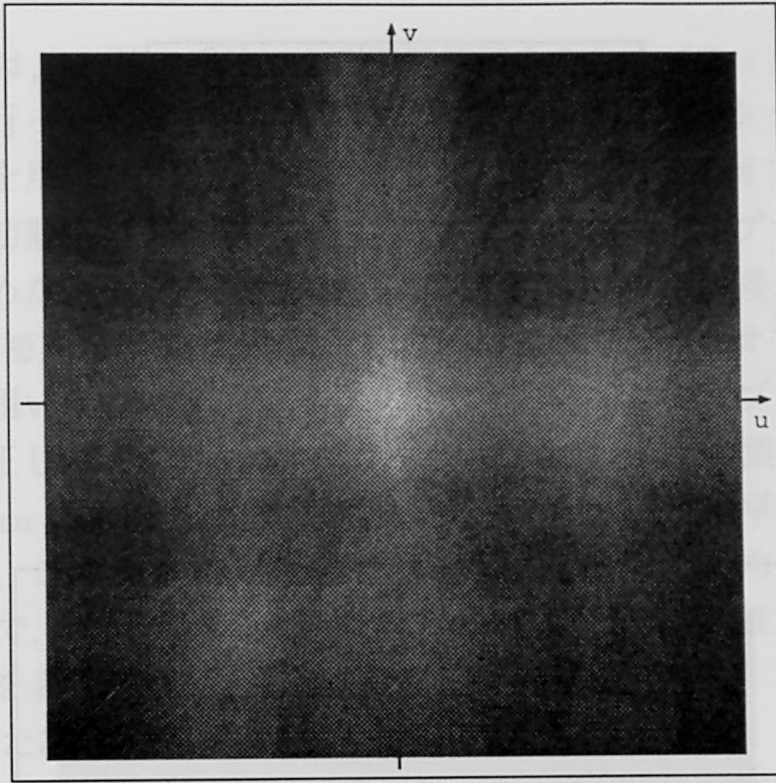


Fig 18: Distribution of Constraint Lines($d = 0.091$)

図 19: 拘束線の分布 ($d = 0.009$)Fig 19: Distribution of Constraint Lines($d = 0.009$)

なかった. また, $d = 0.009$ の時の投票のピークを表 4 に示す. この表からわかるように, 小さな d を用いると, 投票数が少なくなり, ピーク検出の誤差が大きくなる.

以上のことから, 拘束線の変位 d は投票に用いることができる画素数と, オプティカルフローの分解能を考慮して注意深く決定する必要があることがわかる.

極大点 (u, v)	拘束線の数
(0.000, 0.000)	9536
(-0.414, -0.615)	4976
(0.627, 0.045)	4848
(-0.062, 0.537)	3552

表 4: 拘束線の数 of 極大点 ($d = 0.009$)

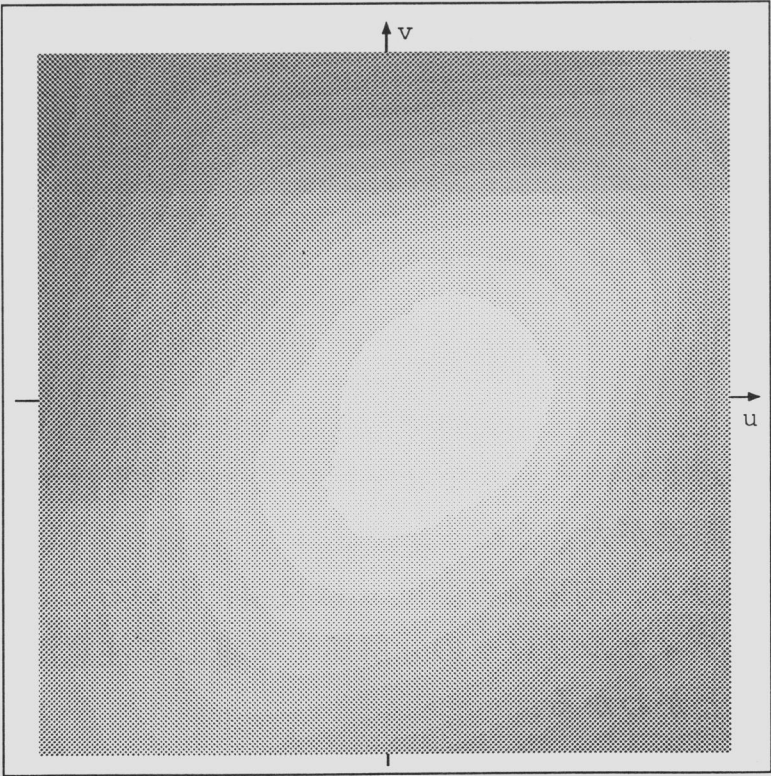


図 20: 拘束線の分布 ($d = 0.9$)

Fig 20: Distribution of Constraint Lines($d = 0.9$)

4.4.2 不連続部を含むオプティカルフローの推定

ここでは、4.2節の手法による不連続部を含むオプティカルフローの推定を行った結果を示す。実験には、前節と同じランダムドットパターンを用いた。この画像に対して、前節で求めたオプティカルフローの初期値を式 (17) における \mathbf{W} として用い、オプティカルフローを求めた。なお、この初期値は、最大で 0.14 の誤差を含む。また、 \mathbf{w} の初期値として、Horn の方法によって求めたオプティカルフローを用いた。なお、Horn の方法によって求めたオプティカルフローは、正しいオプティカルフローから平均で 0.0849 の誤差を持っていた。Horn の方法では、結果的に近傍のデータを加重平均した形になるため、この値は画像に含まれるノイズの影響を平均処理により抑えた上で、領域境界におけるオプティカルフローの推定誤りが伝播した量を表していると言える。

得られた結果をベクトル表示したものを図 21 に示す。この手法によって得られた結果は、正しいオプティカルフローから平均で、0.0288 の誤差を含む。また、正しいオプティカルフローからの変位をベクトル表示したものを図 22 に示す。結果からわかるように、オプティカルフローが誤って推定されている部分は並行移動領域の境界付近に集中している。これは、境界から 3 画素については、中央差分により勾配を正しく計算することができないためであり、勾配法に基づく手法の本質的な限界であるといえる。しかしながら、Horn の方法などのオプティカルフローの連続性を用いる手法のように、境界付近に誤差が広く伝播することは起こっていない。つまり、不連続関数を用いた正則化手法により、境界付近以外について平均を取る形になっており、個々の画素に含まれるノイズによる影響を抑えた結果が得られていると言える。

また、本手法における初期値の影響を調べるため、4.2節の手法に

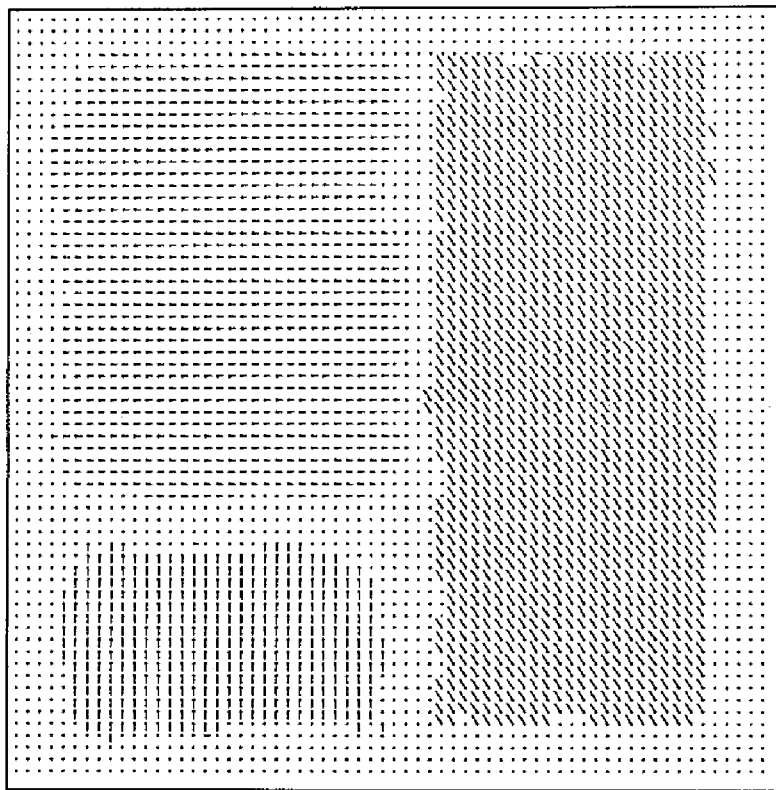


図 21: 本手法により得られたオプティカルフロー

Fig 21: Estimated Optical Flow

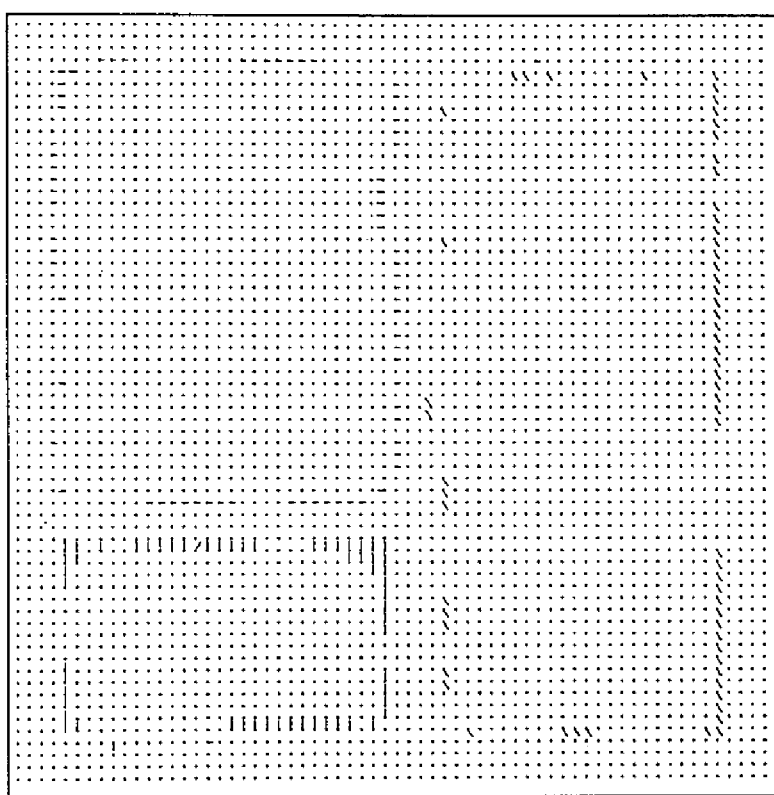


図 22: 本手法により得られたオプティカルフローの誤差

Fig 22: Error of Estimated Optical Flow

よって得られた初期値にノイズを与えたものを用いて同様の実験を行った。用いた初期値は、正しいオプティカルフローとの距離が最大 0.5 になるように、ノイズを加えたものである。得られた結果をベクトル表示したものを図 23 に示す。この手法によって得られた結果は、正しいオプティカルフローから平均で、.0518 の誤差を含む。また、正しいオプティカルフローからの変位をベクトル表示したものを図 24 に示す。この実験では、画像左下の領域に対応するオプティカルフローが大きな誤差を含んでいたため、各画素に対応するオプティカルフローは、他の初期値へと収束していった。このように、正しいオプティカルフローからの距離が、オプティカルフロー間の距離の最小値程度になると、各画素に対応するオプティカルフローは局所最適解に収束する可能性がある。従って、本手法を用いる場合、各領域のオプティカルフロー間の距離の最小値より小さい誤差で初期値を求める必要がある。

4.5 むすび

本章では、不連続部を含む解を解空間の中に含む関数を安定化汎関数として用いる正則化手法を用いたオプティカルフロー推定手法を示した。従来の Tikhonov 型の安定化汎関数を用いた正則化手法では、連続解のみを考えているため、背景の前を移動する物体が存在するような画像に対しては、良い結果が得られなかった。このような問題に対しては、解の不連続性に関して事前に得られる情報をうまく利用できる不連続正則化手法が必要である。本手法では特に、移動する物体は全て並行移動しているという制約を満たすものに対して、正則化手法を適用する手法として、解が離散値を取る安定化汎関数を利用した正則化手法を提案した。また、正則化手法では関数の最小化によって解を得るが、このとき計算に用いる初期値によって解が

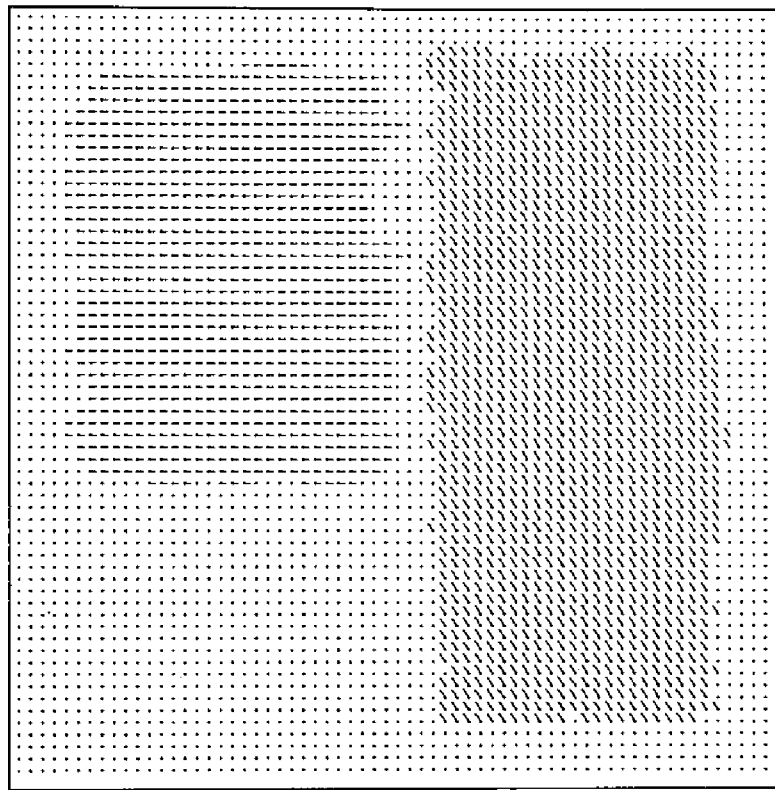


図 23: 誤差の大きな初期値より得られたオプティカルフロー

Fig 23: Estimated Optical Flow from Initial Value with Large Error

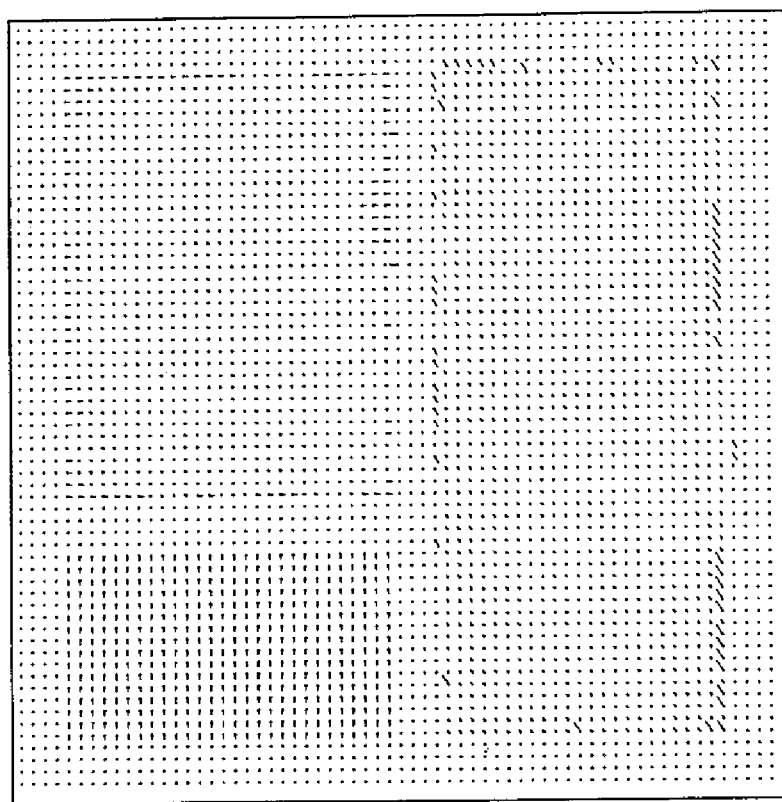


図 24: 誤差の大きな初期値より得られたオプティカルフローの誤差

Fig 24: Error of Estimated Optical Flow from Initial Value with Large Error

変動するという問題がある．従って，良い初期値を得る手法が重要であるが，ここでは，問題の性質を利用して，並行移動している領域のオプティカルフローを求める手法を提案し，これを初期値として利用している．特に，画像に含まれる最高空間周波数成分と量子化 bit 数によって，得られる初期値の誤差が定量的に評価できることを示した．

本手法によって，不連続解を解の中に含む問題に対して，問題の性質によって事前に得られる，不連続に関する情報をうまく利用する安定化汎関数を用いることにより，従来の手法では不連続付近で大きな誤差を含む解しか得られなかったのに対して，良い解が得られることを示した．また，正則化手法における関数の最小化計算において必要になる初期値を，他の手法により求めることにより妥当な解が得られることも示した．

5 章

再帰的分割を用いた適応格子による距離画像からの三次元形状モデルの生成

5.1 まえがき

前章までに述べたように、不連続解に対処する正則化手法では、解の不連続性に対して事前に得られる情報をうまく利用できる不連続正則化手法を用いることが重要である。4章では、解が離散値のみを取る場合について、離散値のみを取る安定化汎関数を用いた不連続正則化手法を提案した。本章では、ノイズを含む距離画像からノイズを除去し、コンピュータグラフィックスで用いる三次元形状モデルを生成する問題に対して、1次元信号の C^1 級の不連続に対して有効な手法である GNC 法を拡張した手法を適用する。

ノイズ除去の問題は、観測されたノイズを含む信号に対して、元の信号に関する事前知識を利用してノイズを除去した信号を推定する問題である。前章までの問題では、得られる観測データと推定すべき情報との間になんらかの操作が含まれていたが、ノイズ除去の問題は観測データと推定すべきデータが同じ情報であるという、正則化手法の応用問題としては最も基本的な問題であるといえる。対象となる物体の表面に不連続部分が存在する場合、前章までと同様の

問題が生じることになる。本章では、この問題に対して GNC 法を適用する手法について検討を行う。また、ノイズを除去したデータに対して、適応的に面パッチを生成するパッチモデル生成法を提案する。

コンピュータグラフィックスでは、表示する対象物の三次元形状モデルを生成しなければならない。従来、三次元形状モデルは人手に頼って入力されていたが、これには非常に手間がかかり多種の形状モデルが必要とされる場合には大きな問題となる。したがって三次元形状モデルの自動生成に対する要求は大きい。

コンピュータグラフィックスで表示する三次元形状モデルとしては、1) リアルタイムでの表示を目的としたもの、2) 精密な表示を必要とするためなるべく誤差の少ないもの、の2種類が考えられる。三次元形状モデルの自動生成を行う場合、これら2種類のモデルを必要に応じて生成できることが必要である。

三次元形状モデルの自動生成の方法として、距離画像から対象物の形状モデルを生成する方法が考えられるが、これはノイズを含む標本点からの曲面再構成の問題と考えられる。

曲面再構成の問題としては以下のようなものがあげられる。Grimson[48]による曲面の再構成では、再構成する曲面の制約条件を設けることにより、少ない標本点を用いて、規則的な格子上の点で表現される曲面の再構成を行っている。この研究では、再構成される曲面は一様な格子上の点で表現されるが、それぞれの格子点を四隅とするような四角形の面パッチによるパッチモデルを生成する場合、面パッチの数が冗長になる。また、Lin ら [49] による曲面の再構成では、対象物に関する複数の断面から三次元形状を再構成している。この方法では、再構成に用いられるデータが断面像でなければならないという制約があり、距離画像に対しては適用できない。

これらに対してコンピュータグラフィックスによる表示を目的と

した曲面の再構成の研究として、田中ら [50] による距離画像からの三次元形状モデルの生成が挙げられる。一般にパッチモデルによる三次元形状モデルを生成する場合、多くの面パッチを使うことにより、より精密な形状モデルを生成することができるが、表示に時間がかかることになり、このバランスをとることが重要になる。この研究では、曲面を表現する面パッチの数を適応的に変化させることにより、面パッチ数と精密さをうまくバランスさせている。そのために均一に分布した三角形パッチをはじめに作り、距離画像との誤差が大きい部分に対して再帰的にその三角形パッチを分割していくという方法を用いている。しかし、この方法では、均一な格子を再分割していくので、格子点は移動せず、均一でない格子を用いる場合に比べ、同じ曲面を表現するのに冗長な数の面パッチを使っているといえる。Vasilescu ら [51] は面パッチ数及びその形状を変化させることにより少ない面パッチ数で複雑な形状を表現する方法を示している。この研究では、格子点はバネで接続されており、それぞれのバネのバネ定数の大きさは対応する距離画像の勾配に比例するように設定する。従って、形状が複雑な部分に面パッチが集まることになる。また、格子点の移動だけでは必要な精度が得られない部分については、面パッチを分割することによって多くの面パッチによって表現されるようになっている。この手法は理論的な根拠が不明確であり、結果の最適性が保証されない、また面パッチの再分割に関して明確な基準がないなどの問題がある。

ここでは、ノイズを含む距離画像からの曲面復元として、2章で紹介した、GNC 法を用いる方法を検討する。また、三次元形状モデルとして複雑な形状を表現しやすいパッチモデルを考え、距離画像から形状モデルを生成する手法を提案する。形状モデルとして、実時間表示するため、あるパッチ数で距離画像との誤差が小さくなるモ

デル, また正確な表示のため, ある距離画像との誤差がある値以下になるモデル, の2種類のモデルを生成できる手法を考える. このため, 曲面を表現する面パッチとして四角形の再帰的分割を用いた適応格子を用い, その配置も変化させる方法を提案する. また, 形状モデルと距離画像との誤差を少なくするように, 各面パッチの形状を変化させ, 再帰的分割するので, 生成される形状モデルはあるパッチ数において, 面パッチの構造及びその配置が最適である.

5.2 GNC 法を用いた距離画像からの曲面復元

距離画像は, 格子上の点の関数値として, 格子面から物体面までの距離が与えられているものである. GNC 法は1次元信号に対する不連続正則化手法であるが, ここでは距離画像に適用するため2次元への拡張を考える.

GNC 法では, 2章でも説明したように以下の式を最小にすることにより解を得る.

$$E = \sum_i ((u_i - d_i)^2 + \lambda^2 (u_i - u_{i-1})^2) + \alpha Z \quad (36)$$

2次元への拡張を考えた場合, 式(36)における右辺第2項の u の傾きが問題になる. そこで, u の代わりに u の傾き, ∇u を用いることを考える. GNC 法では, データの存在する各点と隣合う点との間に式(4)における $g^{(t)}$ という関数を適用することによって, 不連続が存在するかどうかを漸近的に推定するが, この関数の引数を2次元空間上の傾きとした場合, 不連続がどの方向に存在するかを判定できなくなる. 従って, ここでは, 以下のように2方向への連続性を独立に推定する方法を用いる. いま, 与えられた距離画像を $d_{i,j}$ とし, 推定結果を $u_{i,j}$ とすると, 以下の関数を用いることによりノイ

ズを含む距離画像からノイズを除去することができる。

$$F^{(p)} = \sum_{ij} ((u_{i,j} - d_{i,j})^2 + g^{(p)}(u_{i,j} - u_{i-1,j}) + g^{(p)}(u_{i,j} - u_{i,j-1})) \quad (37)$$

ただし、GNC 法では、正規分布に従うノイズを想定しているため、バーストノイズ等の統計的性質の異なるノイズに対しては、対処できないという問題がある。

5.3 三次元形状モデル

5.3.1 三次元形状モデルの表現方法

三次元形状モデルの表現方法には大きく分けて、(1) ソリッドモデル、(2) パッチモデルの二つが挙げられる。

(1) によるモデルは、三次元閉曲面を表す関数を作ることにより、立体を表現する手法で、物体の内側と外側を表現できる、滑らかな曲面が表現できるという利点がある。しかしながら、複雑な形状を関数で表現することは一般に難しく、表現できる形状は比較的単純なものに限られる。また、物体の表面が関数の解曲面として表されているため、表面の座標を得るためには関数の解を計算しなければならないので、表示に時間がかかるという問題がある。

(2) によるモデルは、面パッチと呼ばれる単純な関数で表される小領域を接続することによって三次元形状を表現するもので、多くの面パッチを接続することによって、複雑な形状を表現することができる。一般的には面パッチ数が多ければ、より複雑な形状を表現できる。また、表示に必要な時間は面パッチ数に比例するため、実時間表示のように表示に要する時間が制限されている場合でも、面パッチ数を調整することによって、形状の複雑さと表示に要する時間を調整することができる。

ここでは、複雑な形状の物体に対して、実時間表示できる三次元形状モデルを生成することを考え、表示時間が面パッチの数で制御できる面パッチによる三次元形状モデルを考える。

5.3.2 面パッチの形

パッチモデルで用いられる面パッチの形には、三角形、四角形、それ以上の多角形、またそれぞれの形状も直角二等辺三角形や正方形に固定したものや自由に変化するものなどが考えられる。また、パッチモデルの構造にも、格子状に並ぶものやランダムに並ぶものなど様々なものが考えられる。

本研究では、距離画像として与えられる物体面の形状にあわせて、適応的に面パッチ数及び面パッチの配置を変化させることにより、面パッチ数が少なくかつ、物体面との誤差が小さくなるような形状モデルの面パッチの構造と配置を生成する。このような方法としては、(a) 初期的に少数の面パッチを生成し、これを再帰的に分割していくことによって、ある面パッチ数において物体面との誤差が最小となるような構造と配置を生成する、(b) 自由に分布した点からボロノイ図を作り三角形面パッチを生成するという二つの方法が考えられる。(a)の方法において手続きの繰り返しによる決定的な方法で面パッチの配置を生成すると、解が局所最適解しか得られないという問題があるが、計算方法が比較的簡単であるという利点がある。(b)の方法は最適解が得られるが、最適な点の分布を計算するためには、すべての組合せを計算するしかなく、計算量的に現実的ではない。従って、ここでは(a)の方法を用いる。

(a)の方法では、面パッチの再帰的な分割を行うため、必要なところに必要な数の面パッチを配置出来るようなパッチモデルの構造が必要である。面パッチの分割方法としては、(1) 四角形と三角形の分

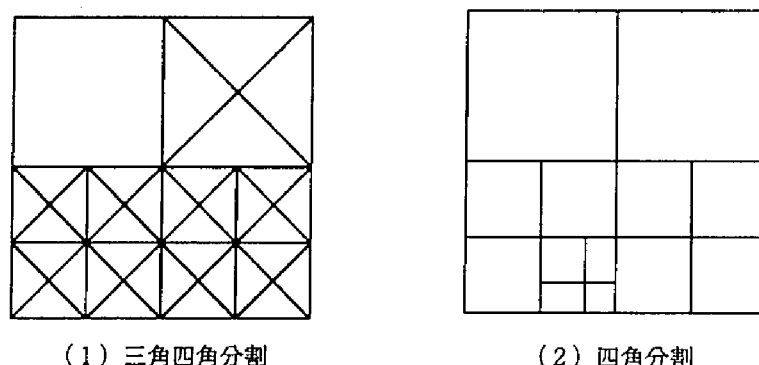


図 25: 面要素の再分割

Fig 25: Re-Dividing Surface Patches

割, (2) 四角形分割などが考えられる. (図 25 参照).

(1) の方法は, ある面パッチを分割すると, 周辺の多くの面パッチも分割する必要があり, 同時に多くの面パッチが増加するという問題がある. これに対して (2) の方法はある面パッチを分割するとき同時に増加する面パッチ数が少ないという利点がある. しかしながら, 基本となる面パッチの形が四角形になるため, 表示装置によっては高速に表示できない場合がある.

本研究では (a) のような方法を用いるため, ある部分の分割によって増加する面パッチ数が少ないことが望ましいことから (2) の四角形の分割を用いる.

この分割方法では 図 26 に示すように 3 辺が交わる点 p で, 曲面の割け目が生じる. そこで, このような三つの面要素が隣接している点 p の高さは, 点 q 及び点 s の線形補間により拘束し, 自由に動く点は点 q や点 s のような四つの面パッチが隣接している点のみとする.

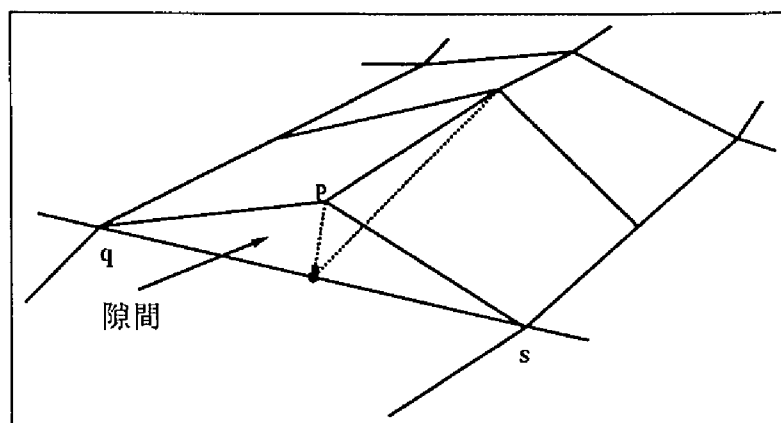


図 26: 割け目の回避

Fig 26: Crack Problem

5.3.3 双一次曲面

面パッチ上の曲面モデルとしては、Coons 曲面、Bezier 曲面、双一次曲面など様々なものがある。ここでは、曲面の計算を 4 頂点の座標のみを用いて行うため、高速な計算を行なうことが出来ること、さらにほとんどの高速グラフィック表示ハードウェアが対応していることから、双一次曲面を用いる。

ある面パッチ s の 4 頂点の座標を (x_0^s, y_0^s, z_0^s) , (x_1^s, y_1^s, z_1^s) , (x_2^s, y_2^s, z_2^s) , (x_3^s, y_3^s, z_3^s) とする。これらの点による双一次曲面 $f^s(x, y)$ は以下のよう表される。

$$\begin{aligned}
 f^s(x, y) = & z_0^s(1-u)(1-w) \\
 & + z_1^s(1-u)w \\
 & + z_2^s u(1-w) \\
 & + z_3^s uw
 \end{aligned} \tag{38}$$

なお、 u , w は以下の式を解くことによって x , y の関数として得ら

れる.

$$\begin{aligned} x = & x_0^s(1-u)(1-w) + x_1^s(1-u)w \\ & + x_2^s u(1-w) + x_3^s uw \end{aligned} \quad (39)$$

$$\begin{aligned} y = & y_0^s(1-u)(1-w) + y_1^s(1-u)w \\ & + y_2^s u(1-w) + y_3^s uw \end{aligned} \quad (40)$$

面パッチの集合 $S = \{s_1, s_2, \dots, s_{N_{patch}}\}$ によって作られる曲面は以下のように表される.

$$f^S(x, y) = \sum_{s \in S} f^s(x, y) \delta^s(x, y) \quad (41)$$

ただし, $\delta^s(x, y)$ は, 面パッチ s の XY 平面への投影領域の中に点 (x, y) が含まれていれば 1, そうでなければ 0 となる関数である (図 27 参照).

5.4 三次元形状モデルの生成

5.4.1 距離画像からの三次元形状モデルの生成

本研究では, 距離画像として与えられる物体面との距離がなるべく小さくなるような曲面を持つ形状モデルを生成する. 面と面との距離としてはいくつかのものが考えられるが, ここでは z 軸に平行な直線と 2 平面の交点の距離の 2 乗和を 2 曲面間の距離とする (図 28 参照).

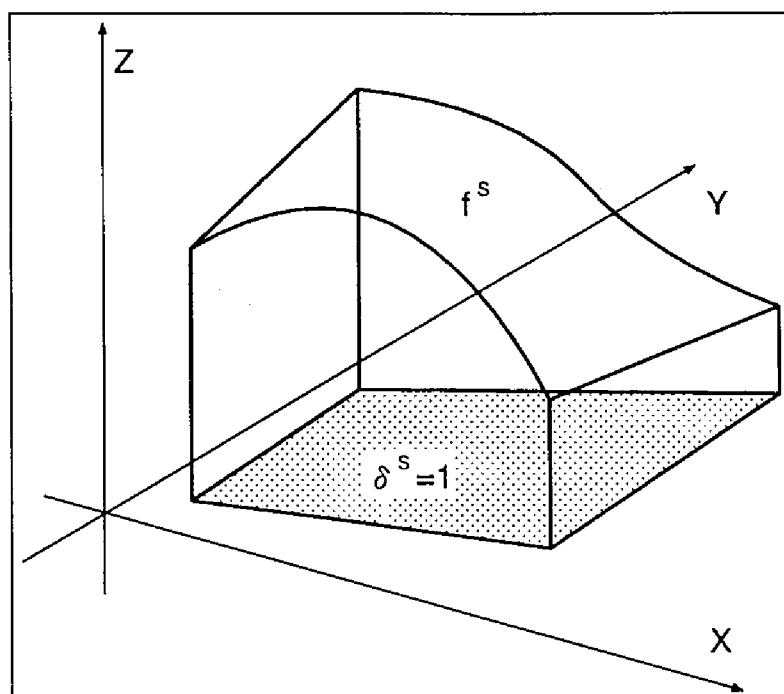


図 27: 面パッチの座標系

Fig 27: Coordinate System of a Surface Patch

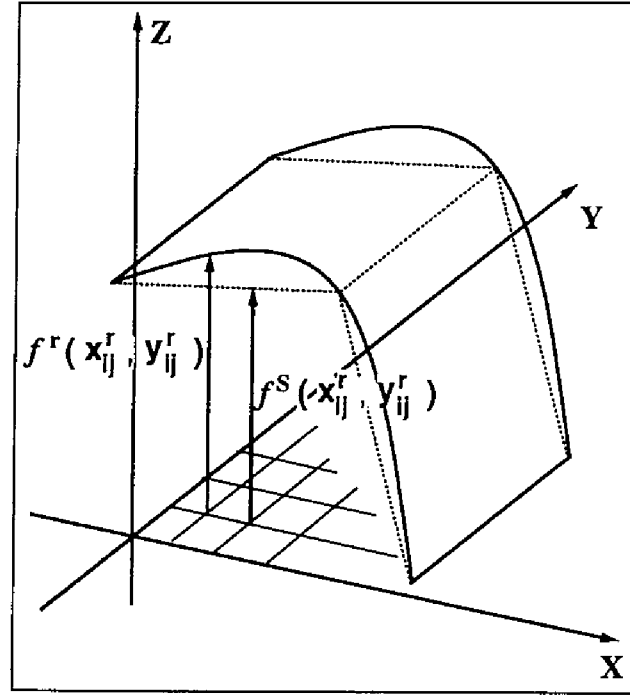


図 28: 距離画像と面パッチの集合

Fig 28: Range Image and the set of patch surface

5.4.2 誤差関数

距離画像を XY 平面上の $N_x \times N_y$ の格子点 (x_{ij}^r, y_{ij}^r) 上の点 $f^r(x_{ij}^r, y_{ij}^r)$, 面パッチの集合 S によって表される曲面を $f^S(x, y)$ とする. すると, 誤差を最小にする面パッチの構造及び配置は, 以下の式を最小にするものであると定義する.

$$E_{error}(x_{ij}^r, y_{ij}^r) = (f^r(x_{ij}^r, y_{ij}^r) - f^S(x_{ij}^r, y_{ij}^r))^2 \quad (42)$$

$$E = \sum_{i,j} E_{error}(x_{ij}^r, y_{ij}^r) \quad (43)$$

この誤差関数の値を以下, 形状モデルの誤差と呼ぶ.

5.4.3 面パッチの形状変化

ここでは、各格子点の移動によって誤差関数を最小化する方法について考察する。この関数の最小化を行うには、式(43)のオイラー方程式を解けばよい。従って、以下の式を反復計算することにより、この関数を最小にする面パッチの頂点 (x_q^s, y_q^s, z_q^s) の値が求まる。

$$\begin{aligned} (x_q^s)^{t+1} &= (x_q^s)^t - \gamma \frac{\partial E}{\partial x_q^s} \\ (y_q^s)^{t+1} &= (y_q^s)^t - \gamma \frac{\partial E}{\partial y_q^s} \\ (z_q^s)^{t+1} &= (z_q^s)^t - \gamma \frac{\partial E}{\partial z_q^s} \end{aligned} \quad (44)$$

なお、 γ は反復計算におけるステップ幅である。
 E の偏微分は以下のように表される。

$$\frac{\partial E}{\partial x_q^s} = \sum_{ij} \left(-2(f^r(x_{ij}^r, y_{ij}^r) - f^s(x_{ij}^r, y_{ij}^r)) \frac{\partial f^s(x_{ij}^r, y_{ij}^r)}{\partial x_q^s} \right) \quad (45)$$

$$\frac{\partial E}{\partial y_q^s} = \sum_{ij} \left(-2(f^r(x_{ij}^r, y_{ij}^r) - f^s(x_{ij}^r, y_{ij}^r)) \frac{\partial f^s(x_{ij}^r, y_{ij}^r)}{\partial y_q^s} \right) \quad (46)$$

$$\frac{\partial E}{\partial z_q^s} = \sum_{ij} \left(-2(f^r(x_{ij}^r, y_{ij}^r) - f^s(x_{ij}^r, y_{ij}^r)) \frac{\partial f^s(x_{ij}^r, y_{ij}^r)}{\partial z_q^s} \right) \quad (47)$$

$f^s(x, y)$ は面パッチである双一次曲面によって定まる関数であるので、式(38), 式(41)より、 $f^s(x, y)$ の偏微分は以下のように表される。

$$\begin{aligned}
& \frac{\partial f^S(x, y)}{\partial x_q^s} \\
&= \sum_{\sigma \in S} \frac{\partial f^\sigma(x, y)}{\partial x_q^s} \delta^\sigma(x, y) \\
&= \sum_{\sigma \in S} \left((z_2^\sigma - z_0^\sigma) \frac{\partial u}{\partial x_q^s} + (z_1^\sigma - z_0^\sigma) \frac{\partial w}{\partial x_q^s} \right. \\
&\quad \left. + (z_0^\sigma - z_1^\sigma - z_2^\sigma + z_3^\sigma) \left(w \frac{\partial u}{\partial x_q^s} + u \frac{\partial w}{\partial x_q^s} \right) \right) \delta^\sigma(x, y)
\end{aligned}$$

$$\begin{aligned}
& \frac{\partial f^S(x, y)}{\partial y_q^s} \\
&= \sum_{\sigma \in S} \frac{\partial f^\sigma(x, y)}{\partial y_q^s} \delta^\sigma(x, y) \\
&= \sum_{\sigma \in S} \left((z_2^\sigma - z_0^\sigma) \frac{\partial u}{\partial y_q^s} + (z_1^\sigma - z_0^\sigma) \frac{\partial w}{\partial y_q^s} \right. \\
&\quad \left. + (z_0^\sigma - z_1^\sigma - z_2^\sigma + z_3^\sigma) \left(w \frac{\partial u}{\partial y_q^s} + u \frac{\partial w}{\partial y_q^s} \right) \right) \delta^\sigma(x, y)
\end{aligned}$$

$$\begin{aligned}
& \frac{\partial f^S(x, y)}{\partial z_q^s} \\
&= \sum_{\sigma \in S} \frac{\partial f^\sigma(x, y)}{\partial z_q^s} \delta^\sigma(x, y) \\
&= \sum_{\sigma \in S} \left((1-u)(1-w) \frac{\partial z_0^\sigma}{\partial z_q^s} + (1-u)w \frac{\partial z_1^\sigma}{\partial z_q^s} \right. \\
&\quad \left. + u(1-w) \frac{\partial z_2^\sigma}{\partial z_q^s} + uw \frac{\partial z_3^\sigma}{\partial z_q^s} \right) \delta^\sigma(x, y)
\end{aligned}$$

ところで, u, w は式 (40) を解いて得られる値である. この式を解くと u, w は x, y に関する 2 次方程式の解として得られることがわかる. これらの式を $\phi(u), \psi(w)$ とする.

$$\phi(u) = a_u u^2 + b_u u + c_u = 0 \quad (48)$$

$$\begin{aligned} a_u &= (x_0^\sigma - x_2^\sigma)(y_1^\sigma - y_3^\sigma) \\ &\quad - (x_1^\sigma - x_3^\sigma)(y_0^\sigma - y_2^\sigma) \end{aligned} \quad (49)$$

$$\begin{aligned} b_u &= (x_0^\sigma - x_2^\sigma)(y_0^\sigma - y_1^\sigma) \\ &\quad + (x_0^\sigma - x)(y_0^\sigma - y_1^\sigma - y_2^\sigma + y_3^\sigma) - \end{aligned} \quad (50)$$

$$\begin{aligned} &(x_0^\sigma - x_1^\sigma)(y_0^\sigma - y_2^\sigma) \\ &\quad - (x_0^\sigma - x_1^\sigma - x_2^\sigma + x_3^\sigma)(y_0^\sigma - y) \end{aligned} \quad (51)$$

$$\begin{aligned} c_u &= (x_0^\sigma - x_1^\sigma)(y_0^\sigma - y) \\ &\quad - (x_0^\sigma - x)(y_0^\sigma - y_1^\sigma) \end{aligned} \quad (52)$$

$$\psi(w) = a_w w^2 + b_w w + c_w = 0 \quad (53)$$

$$\begin{aligned} a_w &= (x_0^\sigma - x_1^\sigma)(y_0^\sigma - y_3^\sigma) \\ &\quad - (x_2^\sigma - x_3^\sigma)(y_0^\sigma - y_1^\sigma) \end{aligned} \quad (54)$$

$$\begin{aligned} b_w &= (x_0^\sigma - x_1^\sigma)(y_0^\sigma - y_2^\sigma) \\ &\quad + (x_0^\sigma - x)(y_0^\sigma - y_1^\sigma - y_2^\sigma + y_3^\sigma) - \\ &\quad (x_0^\sigma - x_2^\sigma)(y_0^\sigma - y_1^\sigma) \\ &\quad - (x_0^\sigma - x_1^\sigma - x_2^\sigma + x_3^\sigma)(y_0^\sigma - y) \end{aligned} \quad (55)$$

$$\begin{aligned} c_w &= (x_0^\sigma - x_2^\sigma)(y_0^\sigma - y) \\ &\quad - (x_0^\sigma - x)(y_0^\sigma - y_2^\sigma) \end{aligned} \quad (56)$$

すると u, w の偏微分は陰関数の性質より, 以下のように表される.

$$\frac{\partial u}{\partial x_q^s} = -\frac{\phi_{x_q^s}}{\phi_u} = -\frac{\phi_{x_q^s}}{2a_u u + b_u} \quad (57)$$

$$\frac{\partial w}{\partial x_q^s} = -\frac{\psi_{x_q^s}}{\psi_w} = -\frac{\psi_{x_q^s}}{2a_w w + b_w} \quad (58)$$

$$\frac{\partial u}{\partial y_q^s} = -\frac{\phi_{y_q^s}}{\phi_u} = -\frac{\phi_{y_q^s}}{2a_u u + b_u} \quad (59)$$

$$\frac{\partial w}{\partial y_q^s} = -\frac{\psi_{y_q^s}}{\psi_w} = -\frac{\psi_{y_q^s}}{2a_w w + b_w} \quad (60)$$

5.4.4 面パッチの再分割

ここでは、四角形の面パッチを再帰的に分割していくことにより、誤差関数を最小化する方法について考察する。

ある面パッチを分割するかどうかは、その面パッチを分割することによって式(43)が減少するかどうかで判断する。しかしながら、分割による関数値の変化は予測しにくいので、ここでは実際に分割を行い、関数値を計算することにより、その値が減少するかどうかで判断する。

面パッチを分割する場合、新たに生成される、四角形の頂点の座標を決定しなければならない。分割する面パッチの4頂点を p_0, p_1, p_2, p_3 とし、新たに生成される頂点を図29のようにする。ここでは、これらの座標を以下のように決定する。

$$p_{01} = \frac{1}{2}(p_0 + p_1) \quad (61)$$

$$p_{02} = \frac{1}{2}(p_0 + p_2) \quad (62)$$

$$p_{13} = \frac{1}{2}(p_1 + p_3) \quad (63)$$

$$p_{23} = \frac{1}{2}(p_2 + p_3) \quad (64)$$

また, p_4 の x, y 座標 (x_4, y_4) は以下のように決める.

$$x_4 = \frac{1}{2}(x_{01} + x_{23}) \quad (65)$$

$$y_4 = \frac{1}{2}(y_{02} + y_{13}) \quad (66)$$

さらに z 座標 z_4 は式 (43) が最小になるように設定する.

5.4.5 面パッチの形状変化及び再分割

本研究では, 三次元形状モデルとして, (1) 実時間表示するため, ある面パッチ数で誤差関数が最小になるもの, (2) 正確な表示のため, 誤差関数がある値以下になるものの2種類の形状モデルの生成を考える.

このような形状モデルを生成するためには, 少ない面パッチ数で誤差関数を小さくするような, 面パッチの構造及びその配置を求めなければならない.

ここでは, 以下のような方法によって面パッチの構造及び配置を求める.

1. 初期的に 4×4 程度の正方格子を生成し, これを初期値とする.
2. ある分割において, 形状変化のみによって誤差関数の値が最小となる面パッチの形状を求める.

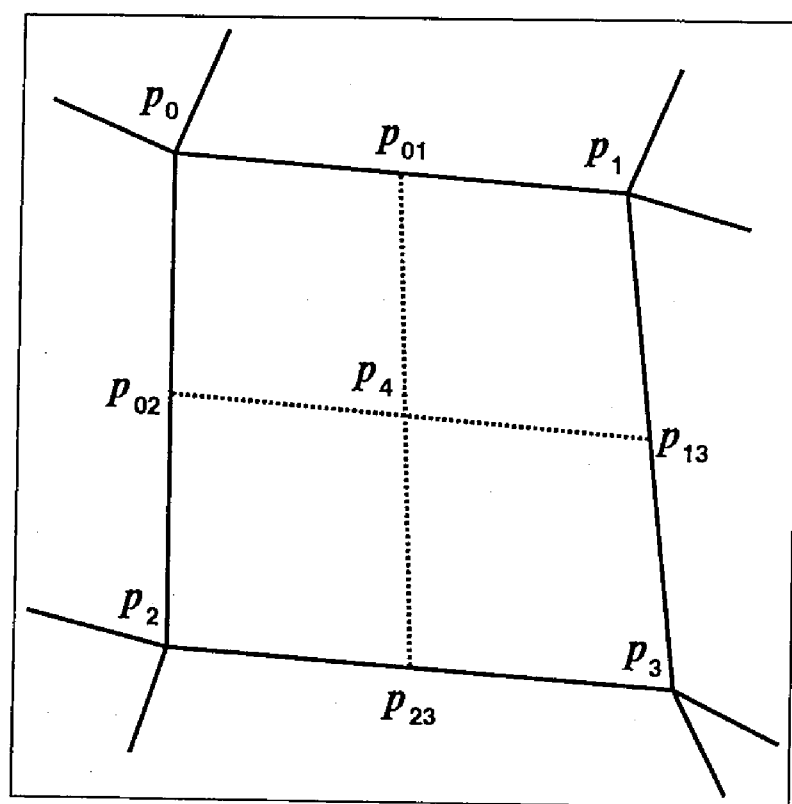


図 29: 分割によって生じる頂点

Fig 29: Apexes created by division of an surface patch

3. 次にその形状において，分割することによって誤差関数の値が減少する面パッチについて分割を行う．
4. 2,3 を面パッチ数が，与えられた面パッチ数になるまで，あるいは誤差関数の値が与えられた値以下になるまで繰り返す．

5.5 実験

ここでは三次元形状モデル生成の実験として，計算機により生成した図 30 のようなノイズの含まれない距離画像を用いる．この距離画像は 128×128 画素，1 画素あたり 256 階調で，以下のような関数によって作られたものである．

$$f(x) = \begin{cases} 0 & (x < 32) \\ \sin\left(\frac{x-32}{128} \times \pi\right) \times 255 & (32 \leq x < 96) \\ 255 & (96 \leq x) \end{cases} \quad (67)$$

また，5.5.3節においては，実距離データに対して三次元形状モデルを生成する．

5.5.1 面パッチの形状変化による形状モデルの生成

面パッチの配置のみを変化させて形状モデルを生成することにより，収束計算の方法の有効性を調べた．

面パッチの初期状態として， 8×8 , 16×16 , 32×32 の正方格子を用いて，図 30 に対して形状モデルを生成した．実験結果を図 31，図 32，図 33 に示す．図にはそれぞれ，面パッチの配置，形状モデルの曲面の立体表示，形状モデルの誤差分布の立体表示，繰り返し計算における誤差の減少過程を示している．なお，繰り返し計算は，誤差の変化率が 10^{-9} 以下になるところまで行った．なお，Vasilescu [51]

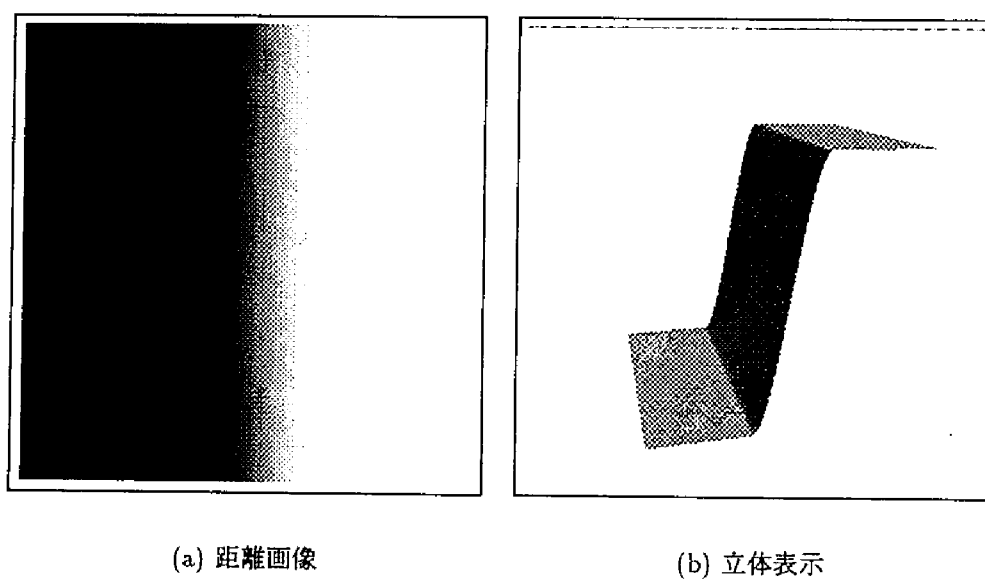


図 30: 原画像

Fig 30: Original Image

	$N_{patch} = 64$	$N_{patch} = 256$	$N_{patch} = 1024$
本手法	7.83996×10^{-5}	5.83717×10^{-6}	5.98154×10^{-7}
バネを用いた手法 [51]	5.28953×10^{-3}	2.22710×10^{-4}	5.35050×10^{-6}

表 5: 形状モデルの誤差

の方法による計算結果における面パッチの配置を、図 34 に示す。また、それぞれの場合の誤差関数の値を表 5 に示す。

結果を見てわかるように、Vasilescu らの方法に比べ、誤差関数の値で 1/10 程度の結果が得られており、本研究における手法が誤差関数の値が小さい形状モデルを生成するという意味で有効な手法であることがわかる。

5.5.2 面パッチの形状変化及びその再分割による形状モデルの生成

面パッチの形状変化、及びその再分割によって、ある誤差以下でパッチ数が最小になる形状モデル、及びあるパッチ数以下で誤差が最小になる形状モデルの生成を行った。

面パッチの初期状態として、 4×4 の正方格子を用いて、図 30 に対して形状モデルを生成した。はじめに、面パッチの数が 64, 256, 1024 になるように形状モデルを生成し、次に形状モデルの誤差がそれぞれ、 7.83996×10^{-5} , 5.83717×10^{-6} , 5.98154×10^{-7} 以下になるように形状モデルを生成した。

このときの結果をそれぞれ、図 35, 図 36, 図 37 及び表 6, 表 7 に示す。図にはそれぞれ、面パッチの配置、形状モデルの曲面の立体表示、形状モデルの誤差分布の立体表示、繰り返し計算における誤差の減少を示した。また、表 6 には、このときの誤差を示した。表 7 には、形状モデルの誤差が、5.5.1 節における実験の誤差以下になる最

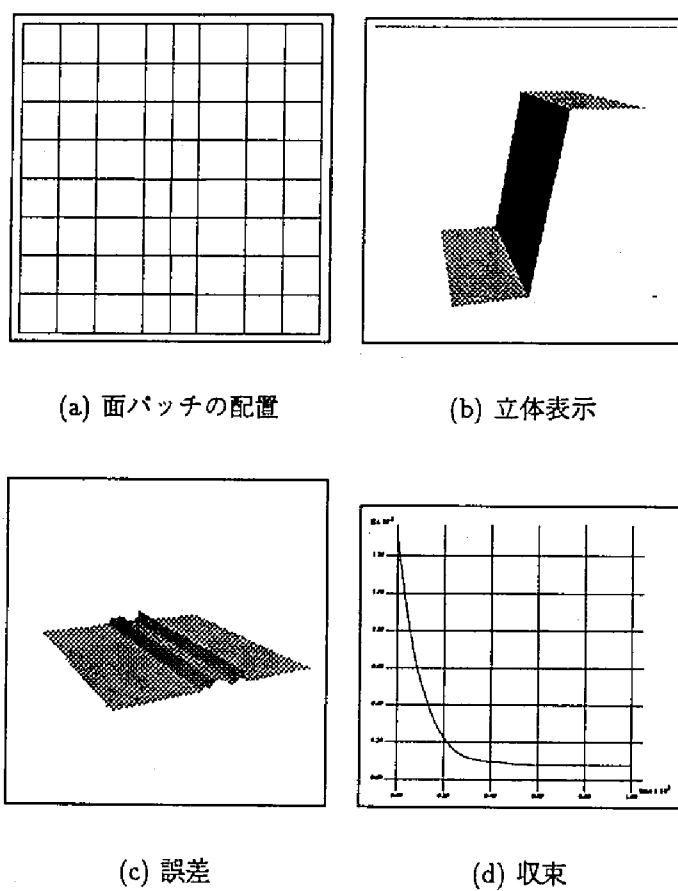


図 31: 形状変化による結果 ($N_{patch} = 64$)

Fig 31: Shape Transformation ($N_{patch} = 64$)

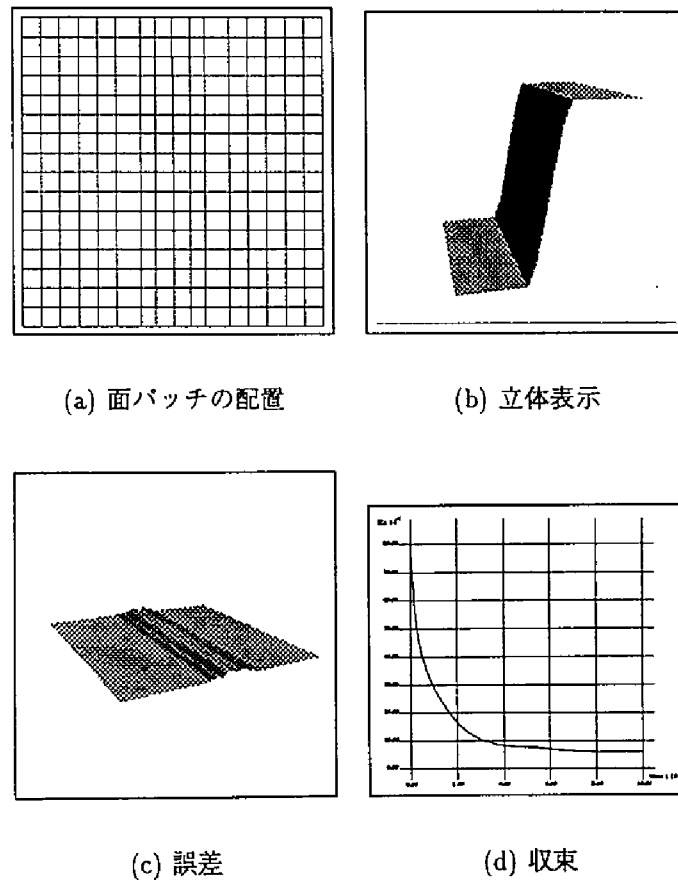


図 32: 形状変化による結果 ($N_{patch} = 256$)

Fig 32: Shape Transformation ($N_{patch} = 256$)

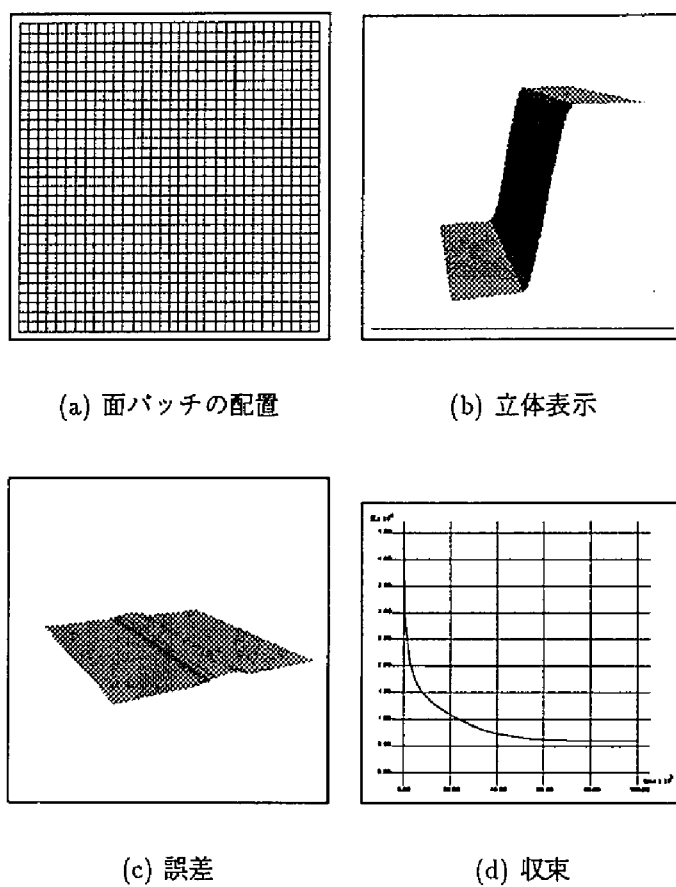
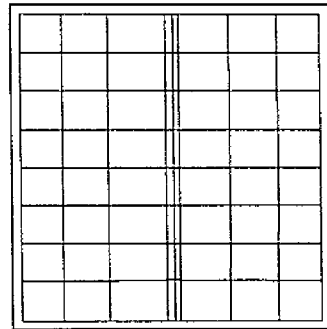
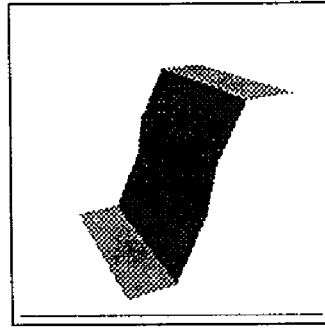


図 33: 形状変化による結果 ($N_{patch} = 1024$)

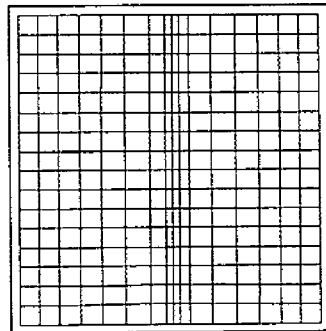
Fig 33: Shape Transformation ($N_{patch} = 1024$)



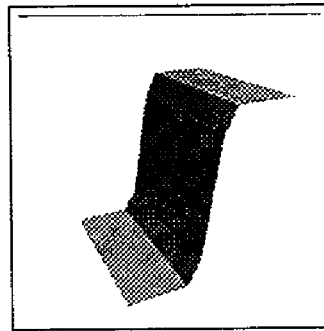
(a) 面パッチの配置
($N_{patch} = 64$)



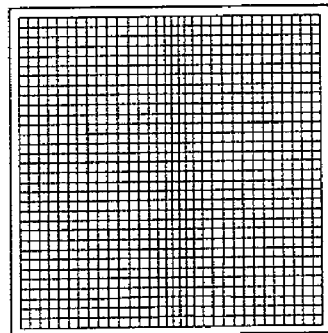
(b) 立体表示 ($N_{patch} = 64$)



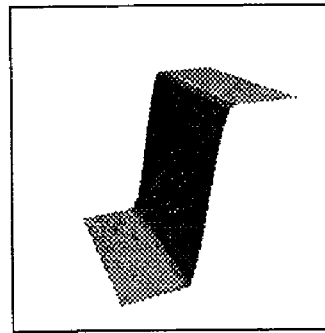
(c) 面パッチの配置
($N_{patch} = 256$)



(d) 立体表示 ($N_{patch} = 256$)



(e) 面パッチの配置
($N_{patch} = 1024$)



(f) 立体表示 ($N_{patch} = 1024$)

図 34: Vasilescu らの手法による結果

Fig 34: Result of method of Vasilescu

5 再帰的分割を用いた適応格子による距離画像からの三次元形状モデルの生成97

$N_{patch} = 64$	$N_{patch} = 256$	$N_{patch} = 1024$
1.2766×10^{-5}	2.27345×10^{-6}	1.61686×10^{-7}

表 6: 形状モデルの誤差

誤差	パッチ数
1.28848×10^{-5}	40
2.72505×10^{-6}	97
3.46637×10^{-7}	823

表 7: 形状モデルの面パッチ数及び誤差

小の面パッチ数における形状モデルの誤差を示した。

実験結果からわかるように、面パッチの形状変化と面パッチの再分割を両方行った場合は、面パッチの形状変化のみを用いて形状モデルを生成した場合と比較して、同じ面パッチ数では誤差の小さい形状モデルかつ、同じ誤差の形状モデルでは、より少ない面パッチ数でモデルを生成できた。

5.5.3 実距離データからの形状モデル生成

ここでは、図 38 に示すような、レーザーレンジセンサーによって得られた 256×240 画素、1 画素あたり 256 階調の距離画像に対して、面パッチの形状変化及びその再分割を用いて形状モデルを生成した。

面パッチの初期状態として、 16×16 の格子を用いて、形状モデルのパッチ数が 256 になるものと、誤差が 1.0×10^{-6} 以下になるものを生成する実験を行った。

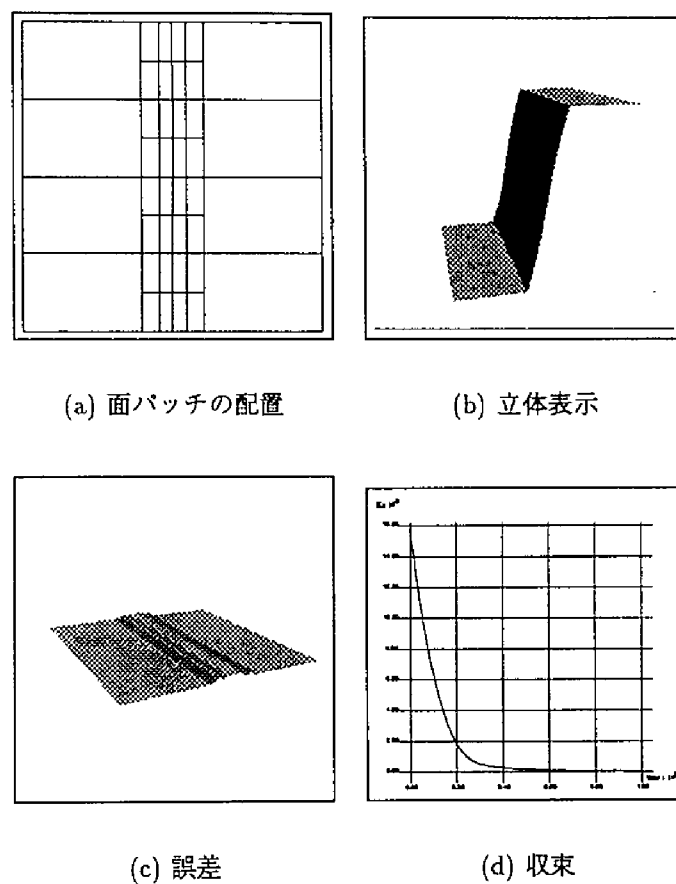


図 35: 形状変化及び分割による結果 ($N_{patch} = 64$)

Fig 35: Shape Transformation and Recursive Division ($N_{patch} = 64$)

5 再帰的分割を用いた適応格子による距離画像からの三次元形状モデルの生成99

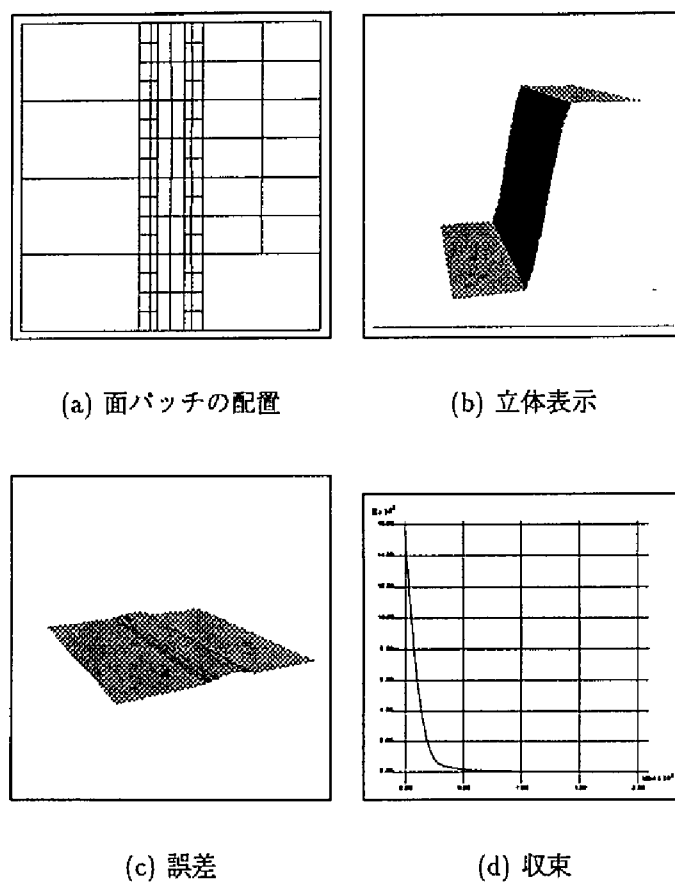


図 36: 形状変化及び分割による結果 ($N_{patch} = 256$)

Fig 36: Shape Transformation and Recursive Division ($N_{patch} = 256$)

5 再帰的分割を用いた適応格子による距離画像からの三次元形状モデルの生成100

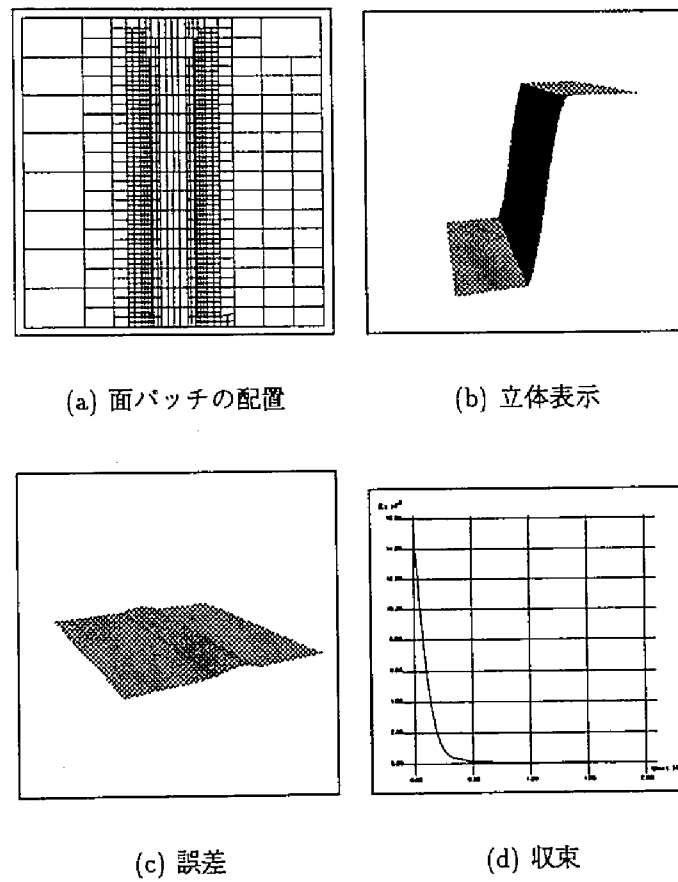


図 37: 形状変化及び分割による結果 ($N_{patch} = 1024$)

Fig 37: Shape Transformation and Recursive Division ($N_{patch} = 1024$)

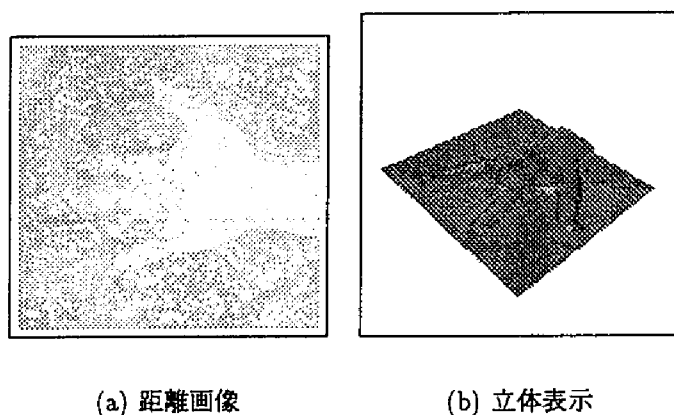


図 38: 原画像

Fig 38: Original Image

誤差	パッチ数
8.84681×10^{-6}	256
9.99897×10^{-7}	1363

表 8: 形状モデルの誤差及び面パッチ数

この時のパッチ数及び誤差を表 8 に示す. また, 誤差 1.0×10^{-6} 以下になる場合の結果を図 39 に示す. 図には, 面パッチの配置, 形状モデルの立体表示を示した. また, このときの面パッチ数は 1363 であった.

実験結果からわかるように原画像の画素数の $1/50$ 程度のパッチ数で, 原画像の形状を表現出来る形状モデルを生成することができた.

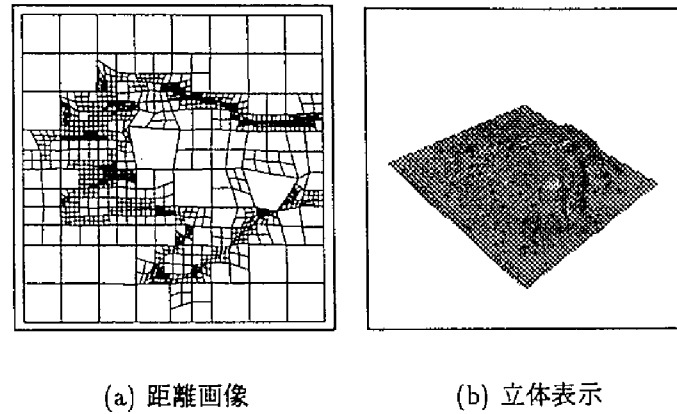


図 39: 実距離画像に対する結果

Fig 39: Resulting Shape Model for Real Range Image

5.6 むすび

本章では、不連続な曲面を持つ対象物の距離データからのノイズ除去及び三次元形状モデルの生成の問題に対して、1次元信号のノイズ除去方法である GNC 法を拡張し、2次元距離画像に対して適用する方法を提案し、さらにノイズを除去された距離画像に対して、適応的に面パッチを配置することによって、最適な形状モデルを生成する手法を提案した。

本手法におけるノイズ除去は、GNC 法を拡張したものであるため、適用できる対象、つまり、対象となる三次元形状の不連続性が明らかであるという特長を持つ。すなわち、不連続性が C^1 級であり、不連続部分の最小の幅が既知である場合でかつ、ノイズが正規分布に従い分散が既知であるならば、ある一定の精度で不連続部分の位置が特定できる。また、三次元形状モデルの生成は、ある面パッチ数の形状モデルとしては、面パッチの構造及び配置が最適になる三次

元形状モデルを生成できる手法である。

本手法により、ノイズが正規分布であり、その分散が既知であるような、 C^1 級の不連続を含む2次元信号に対して、有効なノイズ除去が可能となった。ノイズ除去手法は、与えられたデータに対してパラメトリックな関数を当てはめる問題と等価であるから、初期値が与えられた不良設定問題の解法と等価である。したがって、本手法は2次元情報を対象とした正則化手法に広く応用することが可能である。

6 章

まとめ

初期視覚問題などの不良設定問題に対して，理論的に明らかな根拠を持つ解を与える正則化手法は，有効な求解法である．しかしながら，従来の正則化手法の適用においては，解が連続解でない場合に適用できない，正則化パラメータを経験的に決定しなければならない，関数の最小化問題を解く時の初期値を決めなければならない，という問題があった．本論文では，これらの問題に対して，具体的な応用において，モデルを用いた不連続正則化手法，離散値を取る安定化汎関数を用いた不連続正則化手法，モデルを用いた正則化パラメータの推定法，問題の性質を利用した初期値の計算法を提案した．

第 3 章では，Tikhonov 型の安定化汎関数の問題とパラメータ推定の問題について検討を行った．従来の応用で利用されてきた Tikhonov 型の安定化汎関数を用いたのでは，連続で滑らかな解を持つ問題に対してしか良い解を得ることができず，初期視覚問題で扱う必要性の高い，区分的に滑らかな解を解として含む問題に対しては必ずしも良い解が得られるとは限らない．また，正則化手法では，観測されたデータと推定した解との差を表すペナルティ関数と推定した解の一般的な性質を表す安定化汎関数との比を決める正則化パラメータを決める必要があるが，このパラメータによって解が変化してしま

うため、パラメータを経験的に決定した場合、解の妥当性に関する理論的な根拠が不明確になるという問題がある。これらの問題に対して、初期視覚の問題の中でも特に基本的な処理である、輪郭検出の問題を例として、ノイズを含む画像からも大局的な情報を利用することにより、安定して輪郭を検出することができる、動的輪郭モデル (Snakes) について正則化手法からの検討を行った。従来の Snakes では、滑らかな輪郭しか検出できないという制約があったが、これを解決するため、サンプル輪郭モデルという、ある画像とその画像における輪郭を利用することにより、区分的に滑らかな輪郭も検出出来る手法を提案した。また、Snakes において従来問題とされてきた、パラメータ設定に関して、2枚の画像からパラメータ範囲を推定する手法を提案した。これによって、サンプル輪郭モデルを提示出来る、変形する既知物体の検出などの問題において、安定した輪郭検出を行うことが可能となった。

第4章では、不連続部を含む解を解空間の中に含む関数を安定化汎関数として用いる正則化手法を用いたオプティカルフロー推定手法を示した。従来の Tikhonov 型の安定化汎関数を用いた正則化手法では、連続解のみを考えているため、背景の前を移動する物体が存在するような画像に対しては、良い結果が得られなかった。このような問題に対しては、解の不連続性に関して事前に得られる情報をうまく利用できる不連続正則化手法が必要である。本手法では特に、移動する物体は全て並行移動しているという制約を満たすものに対して、正則化手法を適用する手法として、解が離散値を取る安定化汎関数を利用した正則化手法を提案した。また、正則化手法では関数の最小化によって解を得るが、このとき計算に用いる初期値によって解が変動するという問題がある。従って、良い初期値を得る手法が重要であるが、ここでは、問題の性質を利用して、並行移動している

領域のオプティカルフローを求める手法を提案し、これを初期値として利用している。特に、画像に含まれる最高空間周波数成分と量子化 bit 数によって、得られる初期値の誤差が定量的に評価できることを示した。

本手法によって、不連続解を解の中に含む問題に対して、問題の性質によって事前に得られる、不連続に関する情報をうまく利用する安定化汎関数を用いることにより、従来の手法では不連続付近で大きな誤差を含む解しか得られなかったのに対して、良い解が得られることを示した。また、正則化手法における関数の最小化計算において必要になる初期値を、他の手法により求めることにより妥当な解が得られることも示した。

第5章では、近年需要が増加しているコンピュータグラフィックスにおける三次元形状モデルの自動生成に関して、不連続な曲面を持つ対象物の距離データからノイズ除去を行い、三次元形状モデルを生成する問題に対して、1次元信号のノイズ除去方法である GNC 法を拡張し、2次元距離画像に対して適用する方法を提案し、さらにノイズを除去された距離画像に対して、適応的に面パッチを配置することによって、最適な形状モデルを生成する手法を提案した。

本手法におけるノイズ除去は、GNC 法を拡張したものであるため、適用できる対象、つまり、対象となる三次元形状の不連続性が明らかであるという特長を持つ。すなわち、不連続性が C^1 級であり、不連続部分の最小の幅が既知である場合でかつ、ノイズが正規分布に従い分散が既知であるならば、ある一定の精度で不連続部分の位置が特定できる。また、三次元形状モデルの生成は、ある面パッチ数の形状モデルとしては、面パッチの構造及び配置が最適になる三次元形状モデルを生成できる手法である。

本手法により、ノイズが正規分布であり、その分散が既知である

ような、 C^1 級の不連続を含む2次元信号に対して、有効なノイズ除去手法が可能となった。ノイズ除去手法は、初期値が与えられた不良設定問題の解法と等価であり、本手法は2次元情報を対象とした正則化手法に広く応用することが可能である。

正則化手法は不良設定問題に対して、理論的に明確な根拠を持つ解を与える手法として有効な手法であるが、従来の応用において 1) 不連続解への適用, 2) 正則化パラメータの決定法, 3) 初期値計算法の問題 において問題があった。本論文では、これらの問題に対して、いくつかの解決法を提案した。具体的には、不連続解に対しては、事前に近似解を与える方法、及び離散値を取る安定化汎関数を用いる方法、さらに C^1 級の不連続を検出できる GNC 法を2次元信号に拡張した方法を提案した。また、正則化パラメータの決定法については、サンプルとなる例題を与えることにより、問題の性質を利用してパラメータを決定する方法を提案した。さらに、初期値計算法については、問題の性質を利用して、必要な精度の初期値を得る方法を用いることの有効性を示した。これらの手法により、従来正則化手法が適用しにくかった問題、あるいは適用しても必ずしも良い解が得られなかった問題に対して、適用範囲を広げることが可能となった。より広い範囲の問題に対して、正則化手法が適用できるようになることは、多くの問題に対して、理論的な根拠が明らかな処理手法が実現できることになり、このことは実用的な画像処理手法の実現に必要な方向であると考えられる。

本論文により正則化手法が適用が可能となった画像処理の問題の範囲は大きいと言える。しかしながら、正則化手法を適用できない問題はまだ多く残っている。このような問題として、2次元的な形状を事前知識として利用する正則化手法、弱連続でない不連続性の表現法、局所的に複数の不連続性のモデルの中から一つを選択する機

構などが挙げられる。これらの手法の研究により、より広範囲の問題に対して正則化手法が適用可能になり、より実用的な画像処理手法や信号処理手法が実現可能になると期待される。

参考文献

- [1] D. Marr: "Vision," Freeman, San Francisco, 1982.
- [2] A. N. Tikhonov, V. Y. Arsenin, "Solutions of Ill-Posed Problems," Winston, Washington DC, 1977.
- [3] T. Poggio, V. Torre, C. Koch: "Computational vision and regularization theory," *Nature*, **317**, 6035, pp. 314-319, 1985.
- [4] 谷内田 正彦: "画像処理から画像理解へ", *人工知能学会誌*, 4, 1, pp. 5-11, 1989.
- [5] M. Kass, A. Witkin, D. Terzopoulos: "Snakes: Active Contour Models," *IJCV*, **1**, 3, pp. 321-331, 1988.
- [6] B. K. P. Horn, B. G. Schunck, "Determining Optical Flow", *Artificial Intelligence*, Vol. 17, No. 1, pp. 185-203, 1981.
- [7] K. Ikeuchi, B. K. P. Horn, "Numerical Shape from Shading and Occluding Boundaries," *Artificial Intelligence*, Vol. 17, No. 1, pp.141-184, 1981.
- [8] T. E. Boulton, "What is Regular in Regularization?," 3rd. ICCV, pp.644-651, 1990.

- [9] Demetri Terzopoulos, "Regularization of inverse Visual Problems Involving Discontinuities," Trans. on PAMI, Vol. 8, No. 4, pp.413-424, 1986.
- [10] David Lee, Theo Pavlidis, "One-Dimensional Regularization with Discontinuities," Trans. on PAMI, Vol. 10, No. 6, pp.822-829, 1988.
- [11] S. Geman, D. Geman, "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," Trans. on PAMI, Vol. 6, No. 6, pp.721-741, 1985.
- [12] 志沢 雅彦, "標準正則化理論の多価関数への拡張と多重表面の復元," 信学技法, NC93-34, pp.25 - 32, 1993.
- [13] Andrew Blake and Andrew Zisserman, "Some properties of weak continuity constraints and the GNC algorithm," CVPR, pp. 656-667, 1986.
- [14] Ballard, D.H.: "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognition, vol. 13, no. 2, pp. 111-122, 1981.
- [15] Y.Lamdan and H.J.Wolfson: "Geometric hashing: a general and efficient model-based recognition scheme," Proc. of ICCV, pp.238-249, 1988.
- [16] 上田, 間瀬, 末永: "弾性制約モデルとエネルギー最小化原理による輪郭追跡手法", 信学論, D-II, J75-D-II, 1, pp.111-120, 1992.
- [17] 栄藤, 白井, 浅田: "領域のクラスタリングから得られる混合密度

- 記述による動的輪郭抽出”, 信学技法, PRU91-81, 91, 317, pp.25-32, 1991.
- [18] 天野, 坂口, 美濃, 池田: “Active Contour Model におけるパラメータ決定法”, 1991 年信学秋季全大, D-222, 6-224, 1991.
- [19] 光本, 岡崎: “エネルギー関数とオプティカルフローを用いた口唇輪郭の抽出・補完と追跡”, 情処学論, 31, 3, pp. 444-453, 1990.
- [20] 坂口, 美濃, 池田: “圧力エネルギーを持つ SNAKES”, 1990 年信学春季全大, D-523, 7-275, 1990.
- [21] 坂口, 美濃, 池田: “SNAKES パラメータの設定についての検討”, 信学技報, PRU90-21, 90, 74, pp. 43-50, 1990.
- [22] A.Amini, T.Weymouth, R.Jain: “Using Dynamic Programming for Solving Variational Problems in Vision”, Trans. on Patt. Anal. & Mach. Intell., 12, 9, pp.855-867, 1990.
- [23] A.Amini, S.Tehrani, T. Weymouth: “Using Dynamic Programming for Minimizing the Energy of Active Contours in the Presence of Hard Constraints”, 2nd ICCV, pp. 95-99, 1988.
- [24] L.Westberg, “Hierarchical Contour-Based Segmentation of Dynamic Scenes,” IEEE T-PAMI, vol.14, no.9, pp. 946-952, 1992.
- [25] S.Raghavan, N.Gupta and L.Kanal, “Computing Discontinuity-Preserved Image Flow,” ICPR A, pp. 764-767, 1992.
- [26] K.Y.Wohn, J.Wu and R.W.Brockett, “A Contour-Based Recovery of Image Flow : Iterative Transformation Method,” IEEE T-PAMI, vol 13, no.8, pp. 746-760, 1991.

- [27] J.H.Duncan and T.C.Chou, "On the Detection of Motion and the Computation of Optical Flow," *IEEE T-PAMI*, vol.14, no.3, pp. 346–352, 1992.
- [28] J.F.Boyce, S.R.Protheroe and J.F.Haddon, "A Relaxation Computation of Optic Flow from Spatial and Temporal Cooccurrence Matrices," *ICPR C*, pp. 594–597, 1992.
- [29] J.Konrad and E.Dubois, "A Comparison of Stochastic and Deterministic Solution Methods in Bayesian Estimation of 2-D Motion," *ECCV*, pp. 149–160, 1990.
- [30] C.S.Regazzoni and V.Murino, "Multilevel GMRF-Based Segmentation of Image Sequences," *ICPR B*, pp. 713–716, 1992.
- [31] J.A.Vlontzos and D.Geiger, "A MRF Approach to Optical Flow Estimation," *CVPR*, pp. 853–856, 1992.
- [32] P.Anandan, "A Computational Framework and an Algorithm for the Measurement of Visual Motion," *IJCV*, 2, pp. 283–310, 1989.
- [33] M.Shah, K.Rangarajan and P.S.Tsai, "Generation and Segmentation of Motion Trajectories," *ICPR A*, pp. 74–77, 1992.
- [34] C.Schörr, "Computation of Discontinuous Optical Flow by Domain Decomposition and Shape Optimization," *IJCV*, 8:2, pp. 153–165, 1992.
- [35] R.Agarwal and J.Sklansky, "Estimating Optical Flow from Clustered Trajectories in Velocity-Time," *ICPR A*, pp. 215–219, 1992.

- [36] C.Schörr, "Determining Optical Flow for Irregular Domains by Minimizing Quadratic Functionals of a Certain Class," IJCV, 6:1, pp. 25-38, 1991.
- [37] J.P.Gambotto, "A Region-Based Spatio-Temporal Segmentation Algorithm," ICPR C, pp. 189-192, 1992.
- [38] E.P.Simoncelli, E.H.Adelson and D.J.Heeger, "Probability Distributions of Optical Flow," IEEE C-CVPR, pp. 310-315, 1991.
- [39] J.L.Barron, D.J.Fleet, S.S.Beauchemin and T.A.Burkitt, "Performance of Optical Flow Techniques," IEEE C-CVPR, pp. 236-242, 1992.
- [40] F.Bergholm and S.Carlsson, "A 'Theory' of Optical Flow," CVGIP, vol.53, no.2, pp. 171-188, 1991.
- [41] T.E.Boult and L.G.Brown, "Motion Segmentation Using Singular Value Decomposition," IUW, pp. 495-506, 1992.
- [42] M.Campani and A.Verri, "Motion Analysis from First-Order Properties of Optical Flow," CVGIP, vol.56, no.1, pp. 90-107, 1992.
- [43] 吉本 富士市, 市田 浩三, 清野 武, "区分的 3 次関数を用いた 2 次元データの平滑化の自動的方法," 情報処理, vol. 18, no. 2, pp. 128-134, 1977.
- [44] Andrew Blake, "Comparison of the Efficiency of Deterministic and Stochastic Algorithms for Visual Reconstruction," PAMI, Vol. 11, No. 1, pp. 2-12, 1989.

- [45] M. A. Snyder, "On the Mathematical Foundations of Smoothness Constraints for the Determination of Optical Flow and for Surface Reconstruction," PAMI, Vol. 13, No. 11, pp.1105–1114, 1991.
- [46] M. A. Snyder, "The Mathematical Foundations of Smoothness Constraints: A New Class of Coupled Constraints," IUW, pp.154–161, 1990.
- [47] Demetri Terzopoulos, "The Computation of Visible-Surface Representations," PAMI, Vol. 10, No. 4, pp.417–438, 1988.
- [48] W.E.L.Grimson, "An Implementation of a Computational Theory of Visual Surface Interpolation," CVGIP, 22, pp. 39-69, 1983.
- [49] W.C.Lin, S.Y.Chen, "A New Surface Interpolation Technique for Reconstructing 3D Objects from Serial Cross-Sections," CVGIP, 48, pp. 123-143, 1989.
- [50] 田中 弘美, 岸野 文朗, "3D 曲率を用いた適応的形状復元," 画像の認識・理解シンポジウム予稿集, pp. 397-404, 1992.
- [51] Manuela Vasilescu, Demetri Terzopoulos, "Adaptive Meshes and Shells: Irregular Triangulation, Discontinuities, and Hierarchical Subdivision," CVPR, pp. 829-832, 1992.
- [52] Terrence E. Boult, John R. Kender, "Visual Surface Reconstruction Using Sparse Depth Data," CVPR, pp. 68-76, 1986.
- [53] Terrence E. Boult, John R. Kender, "Visual Surface Reconstruction Using Sparse Depth Data," CVPR, pp. 68-76, 1986.

- [54] Geiger, D., Yuille, A.: "A Common Framework for Image Segmentation", *Int. J. Computer Vision*, vol. 6, no. 3, pp. 227-243, 1991.

謝辞

本論文をまとめるにあたり，京都大学池田克夫教授には，終始的確な御指摘を頂き，研究をすすめることが出来ました．深く感謝の意を表します．

また，京都大学美濃導彦教授には，いつも適切な御指導を頂きました．特に基本的な研究の方針に関して，終始的確な御指摘を頂き，深く感謝いたします．

さらに，研究をすすめるにあたり，池田研究室のみなさんには，いつも議論に加わって頂きました．研究室のみなさんとの議論によって研究は前進いたしました．感謝の意を表します．

関連発表

学会論文誌

1. Shigeru Eiho and Akira Amano: "3-D RECONSTRUCTION OF MYOCARDIUM FROM MRI AND A DISPLAY SYSTEM OF PULSATING 3-D HEART IMAGE," Proc. Computers in Cardiology, pp.211-214, Sep, 1989.
2. 天野 晃, 坂口 嘉之, 美濃 導彦, 池田 克夫: "サンプル輪郭モデルを利用した Snakes," 信学論, vol. J76-D-II, no. 6, pp.1168-1176, 1993.
3. 天野 晃, 坂口 嘉之, 美濃 導彦, 池田 克夫: "再帰的分割を用いた適応格子による距離画像からの三次元形状モデルの生成," 信学論, 掲載予定
4. 天野 晃, 美濃 導彦, 池田 克夫: "不連続正則化手法を用いた並行移動物体のオブティカルフロー推定," 信学論, 投稿中
5. A.Amano and Y.Sakaguchi and M.Minoh and K.Ikeda: "Snakes Using a Sample Contour Model," Proc. of 1st ACCV, pp.538-541, 1993.11.

技術報告

6. 天野 晃, 美濃 導彦, 池田 克夫, 英保 茂, 松田 哲也, 三木 真司, 岡田 幾太郎, 河合 忠一: “Active Contour Model を用いた MRI 断層像からの左心室心壁の追跡,” 信学技法, MBE90-51, vol. 90, no. 151, pp.85-92, 1990. 7.
7. 天野 晃, 坂口 嘉之, 美濃 導彦, 池田 克夫: “サンプル輪郭モデルを利用した Snakes,” 信学技報, PRU91-84, Vol.91, No.317, pp.47-54, 1991. 11.
8. 天野 晃, 坂口 嘉之, 美濃 導彦, 池田 克夫: “正則化を利用した距離画像からの三次元形状モデルの生成,” 信学技報, PRU92-58, Vol.92, No.329, pp.1-8, 1992. 11.
9. 岡本 真明, 天野 晃, 美濃 導彦, 池田 克夫: “垂直撮影画像に同期した水平撮影画像からの歩行者プロファイルの生成,” 計測自動制御学会パターン計測部会, no.23, 1993. 04.
10. 天野 晃, 美濃 導彦, 池田 克夫: “拘束線の分布を利用した並行移動物体のオプティカルフロー推定,” 画像電子学会研究会, 94-04-03, pp.9-12, 1994. 11.
11. 天野 晃: “CVCV-WG 特別報告: コンピュータビジョンにおける技術評論と将来展望 (IV) — 正則化 (Regularization) —,” 情処研報, 95-CV-95, Vol. 95, No. 68, pp.59-66, 1995. 7.

大会発表

11. 天野 晃, 坂口 嘉之, 美濃 導彦, 池田 克夫: “Active Contour

Model におけるパラメータ決定法,” 1991 年信学秋期全大, D-222, 1991. 9.

12. 天野 晃, 美濃 導彦, 池田 克夫: “画像認識における制約充足型画像処理,” 平成 3 年電気連大, S9-4, 1991. 11.
13. 亀田 能成, 天野 晃, 美濃 導彦, 池田 克夫: “手指画像に対するエネルギー関数を用いた三次元モデルマッチング,” 第 43 回情処全大, 7F-2, 1991. 10.
14. 椋木 雅之, 天野 晃, 美濃 導彦, 池田 克夫: “画像認識における記号記述レベル処理での曖昧性対処法の一考察,” 第 43 回情処全大, 7F-5, 1991. 10.